Wrox: Programmer to Programmer

Windows Azure® and ASP.NET MVC Migration

Benjamin Perkins

Join the discussion at p2p.wrox.com

wrox™
A Wiley Brand

www.it-ebooks.info

www.it-ebooks.info

WINDOWS AZURE® AND ASP.NET MVC MIGRATION

INTRODUCTION xiii

▶ **PART I MIGRATION**

CHAPTER 1 Migrating from ASP.NET to ASP.NET MVC 4 2

CHAPTER 2 Creating the ASP.NET MVC 4 Project 25

▶ **PART II ENHANCING**

CHAPTER 3 Understanding ASP.NET MVC 4 Performance Optimization Techniques 71

CHAPTER 4 Fine-tuning the ASP.NET MVC 4 Project for Performance 83

▶ **PART III DEPLOYMENT**

CHAPTER 5 Discussing ASP.NET MVC 4 Windows Azure Deployment Techniques 119

CHAPTER 6 Deploying an ASP.NET MVC 4 Project to Windows Azure 144

▶ **PART IV MONITORING AND TROUBLESHOOTING**

CHAPTER 7 Maintaining an ASP.NET MVC 4 Deployment on Windows Azure 186

CHAPTER 8 Monitoring and Supporting an ASP.NET MVC 4 Project on Windows Azure 204

www.it-ebooks.info

Windows Azure® and ASP.NET MVC Migration

www.it-ebooks.info

Windows Azure® and ASP.NET MVC Migration

Benjamin Perkins

www.it-ebooks.info

Windows Azure® and ASP.NET MVC Migration

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2013 by John Wiley & Sons, Inc., Indianapolis, Indiana

ISBN: 978-1-118-67858-9 (ebk)
ISBN: 978-1-118-74987-6 (ebk)

Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Trademarks: Wiley, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Windows Azure is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

ACQUISITIONS EDITOR
Mary James

PROJECT EDITOR
Maureen Spears

TECHNICAL EDITOR
Don Reamey

PRODUCTION EDITOR
Daniel Scribner

COPY EDITOR
San Dee Phillips

EDITORIAL MANAGER
Mary Beth Wakefield

FREELANCE EDITORIAL MANAGER
Rosemarie Graham

ASSOCIATE DIRECTOR OF MARKETING
David Mayhew

MARKETING MANAGER
Ashley Zurcher

VICE PRESIDENT AND EXECUTIVE GROUP PUBLISHER
Richard Swadley

VICE PRESIDENT AND EXECUTIVE PUBLISHER
Jim Minatel

PROOFREADER
James Saturnio, Word One

COVER DESIGNER
Ryan Sneed

www.it-ebooks.info

ABOUT THE AUTHOR

BENJAMIN PERKINS (MBA, MCSD.Net, ITIL Management) is currently employed at Microsoft in Munich, Germany, as a senior support escalation engineer on the IIS and ASP.NET team. He has been working professionally in the IT industry for almost two decades. He started computer programming with QBasic at the age of 11 on an Atari 1200XL desktop computer. He takes pleasure in the challenges that trouble-shooting technical issues have to offer and savors in the rewards of a well-written program. After completing high school he joined the United States Army and served as an M1A3 tank driver in the Gulf War of 1990. After successfully completing his military service, he attended Texas A&M University in College Station, Texas, where he received a Bachelor of Business Administration degree in management information systems.

His roles in the IT industry have spanned the entire spectrum from programmer, to system architect, technical support engineer, to team leader and mid-level management. While employed at Hewlett-Packard, he received numerous awards, degrees, and certifications. He has a passion for technology and customer service, and looks forward to trouble shooting and writing more world-class technical solutions.

"My approach is to write code with support in mind, and to write it once correctly and completely so we do not have to come back to it again, except to enhance it."

Benjamin is married to Andrea and has two wonderful children, Lea and Noa.

ABOUT THE TECHNICAL EDITOR

DON REAMEY is an Architect / Principal Engineer for TIBCO Software working on TIBCO Spotfire business intelligence analytics software. Prior to TIBCO, Don spent 12 years with Microsoft corporation as a Software Development Engineer working on SharePoint, SharePoint Online, and InfoPath Forms Service. Don has also spent 10 years writing software in financial service industry for capital markets.

www.it-ebooks.info

ACKNOWLEDGMENTS

I WOULD LIKE TO THANK the editors at Wiley Publishing who helped get the content of this book into great shape for the reader. Without them, this book would not have been possible.

www.it-ebooks.info

CONTENTS

INTRODUCTION TO THE WINDOWS AZURE BOOK SERIES *xiii*

INTRODUCTION *xv*

PART I: MIGRATION 1

CHAPTER1: MIGRATING FROM ASP.NET TO ASP.NET MVC 4 2

Getting Started 3
Comparing ASP.NET to ASP.NET MVC 3
Understanding the Model-View-Controller 6
Using Master Pages versus Shared Views 7
When to Use User Controls versus Partial Views 8
Understanding the Statelessness of ASP.NET MVC 8
Understanding Strongly Typed Class References 9
ASP.NET, ASP.NET MVC, or Both? 9
What Are the Differences Between ADO.NET and NHibernate? 10
Understanding the Data Access Layer 10
Understanding the Object Relational Mapping 12
Understanding Mapping by Code 14
Accessing Data from the Database 15
Examining Innovations from IIS 6 to IIS 7.0/7.5/8 16
Introducing Team Foundation Server 19
Using Test Driven Development Within ASP.NET MVC 22
Summary 24

CHAPTER 2: CREATING THE ASP.NET MVC 4 PROJECT 25

Changing the Look and Feel of Your Website 26
Creating the Blog List from an XML RSS File 29
Adding the BlogList Class to the Models Directory 30
Adding Methods to the HomeControllers File 31

www.it-ebooks.info

Modifying the Index() Method 33
Displaying the List of Blogs in the View 34
Adding the Html.ActionLinks 35
Creating a Local Test Database 38
Downloading and Installing SQL Server 38
Creating a New SQL Server Database 39
Implementing NHibernate into an ASP.NET MVC 4 Application 40
Creating the NHibernate Classes and Configuration 42
Creating and Adding the BlogNavBar Partial View 49
Adding a Partial View 49
Adding Static Content to the View 50
Creating a Sample View and Adding the _BlogNavBar 50
Adding Dynamic Content to the _BlogNavBar Partial View 52
Creating the Archive List Web Page 59
Creating the ArchiveList.cshtml View 59
Adding the ArchiveList() Action Result Method 60
Create and Implement a Custom MapRoute 60
Retrieving the Archive Blog Data with LINQ to NHibernate 61
Modifying the Views to Display and Link to Data 63
Migrate a Blog Entry from ASP.NET with Feedback Form and Comment List 64
Adding a Controller to Manage Requests to All Blogs 65
Creating a Method to Retrieve Blog Details 65
Updating the Example Blog Link 66
Creating a Shared Partial View for the Blog 67
Adding Partial View to the Blog 68
Summary 68

PART II: ENHANCING 70

CHAPTER 3: UNDERSTANDING ASP.NET MVC 4 PERFORMANCE OPTIMIZATION TECHNIQUES 71

Setting a Performance Baseline 72
Using Online Tools for Performance Testing and Optimization Tips 73
Understanding Bundling and Minification 74
Scaling a Windows Azure Cloud Service 77

www.it-ebooks.info

Fifteen Performance Enhancing Tips 79
Useful Links 81
Summary 82

CHAPTER 4: FINE-TUNING THE ASP.NET MVC 4 PROJECT FOR PERFORMANCE 83

Using Fiddler to Capture Performance Statistics 84
Implementing MiniProfiler 88
Implementing into ASP.NET Website 89
Implementing into the ASP.NET MVC 4 Web Role 95
Capturing Performance Data with IE F12 Developer Tools 99
Employing Google PageSpeed — ASP.NET Website 100
Leveraging Browser Caching 101
Enabling Compression 102
Serve Scaled Images 103
Optimizing Images 104
Bundling and Minifying JavaScript and CSS 106
Understanding the Impact of Bundling and Minifying Files 107
Implementing Bundling 107
Implementing Minification 108
Configuring Compression and Caching 110
Implementing Compression 110
Changing the Output Caching 113
Comparing ASP.NET MVC 4 Performance After Tuning 115
Summary 117

PART III: DEPLOYMENT 118

CHAPTER 5: DISCUSSING ASP.NET MVC 4 WINDOWS AZURE DEPLOYMENT TECHNIQUES 119

Preparing Your Application for the Windows Azure Platform 120
Straightforward Implementation 120
Scalability, Availability, and Durability 120
Releasing Internal Resources 120
Quality Support/Experienced Practitioners 121
Mobility 121
Reduced Cost 121

www.it-ebooks.info

Understanding Developer Centers and Supported SDKs 121
Introducing Cloud Computing Services 122
Understanding Cloud Computing Service Models 123
Understanding Deployment Models 124
Understanding Private and Public Clouds 125
Understanding a Community Cloud 127
Understanding a Hybrid Cloud 128
Accessing the Windows Azure Platform 128
Choosing Your Windows Azure Services 131
Using Azure Web Site versus Cloud versus Virtual Machine 131
Understanding Data Storage Features 132
What is Windows Azure SQL Reporting? 133
What is HDInsight? 133
Using Active Directory 133
Understanding Deployment Options 134
Integrating Source Control with a Cloud Service 134
Integrating Source Control with an Azure Web Site 135
Deploying Web Roles 138
Planning Database Migration and Storage 140
Monitoring the Status of a Deployment 141
Summary 143

CHAPTER 6: DEPLOYING AN ASP.NET MVC 4 PROJECT TO WINDOWS AZURE 144

Accessing Windows Azure 145
Creating the Window Azure Web Site and Cloud Service 146
Creating a Website 146
Creating a Cloud Service 148
Adding and Connecting a SQL Database 150
Adding a SQL Server Database 150
Connecting to the Database 153
Deploying and Testing Your Code 156
Converting an ASP.NET MVC 4 Project to a Cloud Service 157
Deploying with Visual Studio Publishing Features 162
Setting a TFS Connection and Publishing the ASP.NET MVC 4 Website 170
Accessing the Team Foundation Server 170
Adding Links Between TFS and Windows Azure 175

www.it-ebooks.info

Connecting a Windows Azure Web Site to a GitHub Code Repository 178
Publishing an ASP.NET MVC 4 Website Using FTP 182
Setting Up FTP Capability 183
Publishing the Project 184
Summary 184

PART IV: MONITORING AND TROUBLESHOOTING 185

CHAPTER 7: MAINTAINING AN ASP.NET MVC 4 DEPLOYMENT ON WINDOWS AZURE 186

Monitoring a Windows Azure Web Site 187
Monitoring with the Dashboard 187
Monitoring with the Website's Management Console 188
Monitoring a Windows Azure Cloud Service 191
Using the Task Manager and Event Viewer 191
Using IIS and PowerShell 191
Using the Cloud Service Management Console 192
Configure a Cloud Service from Visual Studio 196
Monitor a Cloud Service from within Visual Studio 198
Management and Monitoring Tools for Windows Azure 199
Open Source Tools 199
Windows Azure Management API 200
Windows Azure PowerShell Cmdlets 201
Microsoft Tools for Monitoring and Managing Windows Azure 202
Summary 203

CHAPTER 8: MONITORING AND SUPPORTING AN ASP.NET MVC 4 PROJECT ON WINDOWS AZURE 204

Monitoring and Supporting an ASP.NET MVC 4 Web Site on Windows Azure 204
Accessing the Graph and Usage Overview 205
Adding Metrics to the List of Monitored Attributes 206
Configuring Diagnostics for the Website 207
Downloading and Analyzing Diagnostic Logs 210
Streaming Diagnostic Logs 213

www.it-ebooks.info

Monitoring and Supporting an ASP.NET MVC 4 Cloud Service on Windows Azure **216**
Updating Diagnostic Settings on a Live Service 216
Configuring a Remote Desktop Connection 218
Setting Up a Remote Desktop Connection for an Existing Cloud Service 224
Viewing the Cloud Service Usage Dashboard 237
Summary **241**

www.it-ebooks.info

INTRODUCTION TO THE WINDOWS AZURE BOOK SERIES

It has been fascinating watching the maturation of Windows Azure since its introduction in 2008. When it was announced, Azure was touted as being Microsoft's "new operating system." And at that level, it has not really lived up to its billing. However, if you consider Azure to be a collection of platforms and tools that allow you to cloud-enable your corporation's applications and infrastructure, well, now you're on the right track.

And, as it turns out, a collection of co-operating tools and services is the best way to think of Azure. The different components that comprise Azure become the building blocks that allow you to construct an environment to suit your needs. Want to be able to host a simple website? Well, then Azure Web Sites fits the bill. Want to move some of your infrastructure to the cloud while leaving other systems on premise? Azure Virtual Networking gives you the capability to extend your corporate domain to include machines hosted in Azure. Almost without exception, each twist and turn in your infrastructure roadmap can take advantage of the building blocks that make up Windows Azure.

A single book covering everything that encompasses Azure would be huge. And, because of the breadth of components in Azure, such a book is likely to contain information that you are not necessary interested in. For this reason, the Windows Azure series from Wrox takes the same "building block" approach that Azure does. Each book in the series drills deeply into one technology. If you want to learn everything you need to work with a particular technology, then you could not do better than to pick up the book for that topic. But you don't have to dig through 2,000 pages to find the 120 pages that matter to you. Each book stands on its own. You can pick up the books for the topics you care about and know that's all that you will get. And you can leave the other books until desire or circumstance makes them of interest to you.

So enjoy this book. It will give you the information you need to put Windows Azure to use for you. But as you continue to look to other Azure components to add to your infrastructure, don't forget to check out the other books in the series to see what topics might be helpful. The books in the series are:

- *Windows Azure and ASP.NET MVC Migration* by Benjamin Perkins, Senior Support Escalation Engineer, Microsoft
- *Windows Azure Mobile Services* by Bruce Johnson, MVP, Partner, ObjectSharp Consulting
- *Windows Azure Web Sites* by James Chambers, Product & Community Development Manager, LogiSense

www.it-ebooks.info

- *Windows Azure Data Storage* by Simon Hart, Software Architect, Microsoft
- *Windows Azure Hybrid Cloud* by Danny Garber, Windows Azure Solution Architect, Microsoft; Jamal Malik, Business Solution Architect; and Adam Fazio, Solution Architect, Microsoft

Each one of these books was written with the same thought in mind: to provide deep knowledge of that one topic. As you go further into Azure, you can pick and choose what makes sense for you from the other books that are available. Constructing your knowledge using these books is like building blocks. Which is just the same manner that Azure was designed.

Bruce Johnson
Azure Book Series Editor

www.it-ebooks.info

INTRODUCTION

If you haven't already started planning your migration from a Windows Server 2003 environment, then I recommend you get started. As of July 14, 2015, extended support for this operating system and the associated components (that is, IIS 6) will stop. This means that Microsoft will no longer create security updates or take support calls for this version.

Whether you're almost finished with your migration or just beginning, you shouldn't underestimate the importance of choosing your next platform. You may want to upgrade to the most current versions of all the components necessary to run your system while keeping the number of required changes to a minimum. You might even want to get venturesome and convert some of your ADO.NET queries to use LINQ, implement an ORM, or migrate from Oracle to SQL Server.

You have many reasons to upgrade to the newest operating system:

- It's a good opportunity to not only change platforms, but also to reinvent your system.
- You can look at your system from an opportunity perspective instead of from a cost perspective.
- You can find fresh innovations now available on these new platforms, and gain additional customers while retaining your key talent.
- In addition to innovating, you can take a closer look at support and maintenance cost reduction opportunities on these new platforms. You may get a pleasant surprise.

As exciting as this may sound, you can anticipate significant effort and some risk when you move to an entirely new platform. You should make these changes in phases; for example, deploy the system "as-is" to the platform and then migrate to the new framework or DBMS system one after another. It is recommended that you make as few changes as possible at a time making it easier to troubleshoot any problems that arise. Understanding the requirements of your system and the capabilities of the new environment are key factors in a successful migration.

WHO THIS BOOK IS FOR

This book is for technology professionals looking to take advantage of the functionality that ASP.NET MVC 4 and Windows Azure Cloud Services offers. You should already have a good understanding of ASP.NET, ADO.NET, NHibernate, and IIS.

www.it-ebooks.info

WHAT THIS BOOK COVERS

This book explains some of these new capabilities and walks you through a real-world migration of an ASP.NET 2.0/ADO.NET website hosted on Windows Server 2003 using IIS 6 to an ASP.NET MVC/ORM (NHibernate) website-hosted on Windows Azure.

HOW THIS BOOK IS STRUCTURED

This book is organized into four sections:

- Migration
- Enhancing
- Deployment
- Monitoring and Troubleshooting

Each section contains two chapters. The first chapter of each section contains concepts you need to understand about the migration process. These include, for example, what you need to know to migrate from ADO.NET to an ORM like NHibernate, to understand the Entity Framework, or to determine the major differences between ASP.NET and ASP.NET MVC. The second chapter contains detailed exercises of the most interesting modifications made between the old and new versions of the technology. These step-by-step instructions are intended to give you a deeper understanding of the migration process so you can reproduce the migration, if required.

If, while performing steps in an exercise chapter, you find that you don't understand the reasoning behind them, you can always click the section's hyperlink, attached to the exercise heading, to access the appropriate discussion in concept chapter.

Migration

Rather than use “Migration” for the name of this section, “Transformation” might have been used. The differences between ASP.NET and ASP.NET MVC are big enough to say that this is not only a change in technology, but also a change in some fundamental concepts that impact design, testing, and maintenance decisions. The way you previously worked with web forms in those capacities must be tweaked to fit into this new technical framework. Not only do you learn about those differences in this section, but you also discover how to migrate from ADO.NET to NHibernate, Team Foundation Server, and Test Driven Development, all of which are conceptual changes if you haven’t previously used them.

In addition, an introduction to Windows Azure and some of the concepts features available are discussed in this section. This introduction prepares you for the deeper analysis covered in the chapters that follow.

www.it-ebooks.info

Enhancing

Performance matters a great deal. For systems that get millions of hits per day, a system can gain a lot of opportunity even when the improvement is less than one-half or one-quarter of a second. For example, if a Facebook page responded 500 ms faster than it does now, this would result in a three percent increase in traffic.

The more recent versions of ASP.NET and the .NET Framework have seen some significant advancement in the available features for optimization, for example, minification, bundling, compression, and caching. Using tools such as ACT, Google PageSpeed, or F12 Developer Tool Suite that measure the speed of your system and provide some tips on how to improve performance are valuable. A test website, available for download on the Wrox website here `www.wrox.com/go/azureaspmvcmigration`, serves as an example for testing these tools and viewing the results of implementing the author's suggestions. The sample ASP.NET website is located at `http://aspnet.thebestcsharpprogrammerintheworld.com`; the sample ASP.NET MVC 4 Windows Azure Web Site (currently in Preview mode, that is, beta) is located here at `http://mvc-4.azurewebsites.net`; and the sample ASP.NET MVC 4 Windows Azure Web Role is located at `http://mvc-4.cloudapp.net`.

> **WARNING** *Windows Azure Web Sites are currently in Preview mode. All references to this feature are subject to change.*

Finally, some tips and discussion about how to scale on a Windows Azure program are provided.

Deployment

Moving your source code between your different testing environments requires a clearly defined process and, in many cases, a partially or fully dedicated Release Manager. Keeping the database connections, resource dependencies, source code, operating system version/patches, system configuration, and so on identical between environments can be a daunting task. It is a daunting task because to do it well the Release Manager must make sure system testing, integration testing, and production servers are identical so that results from testing are the same on all environments. If there are differences between servers, the behavior of the system may be different.

One of the benefits of the Windows Azure Cloud Service is its Staging and Production environments, which enable you to deploy your source code to the cloud and test it without impacting or changing your production/live system. This a very valuable benefit, and reduces the risks that come with making changes to a production system. When testing is complete, you can switch your Staging environment to become the Production instance on the Widows Azure platform.

www.it-ebooks.info

You have many ways to move your code from a local location to a website or Cloud Service hosted on the Windows Azure platform. For example, you can link your website or Cloud Service directly to Team Foundation Server by selecting the Set Up TFS Publishing link on the Dashboard for the given website or Cloud Service. You can also publish from within Visual Studio, using Web Deploy, or by using an FTP application such as FileZilla or WS_FTP.

Monitoring and Troubleshooting

Just because business owners signed off on your website and it is deployed to a live environment, that doesn't mean your work is complete. At this point, some of the most complex and challenging aspects of technology start. There is no tool nor person who can test or code for every user behavior and action occurring within your application. Therefore, you need to be ready to troubleshoot and resolve issues for which you have not planned. For example, the following happens only in the Production environment: The performance of a transaction is very slow, logging in to the system results in a timeout or the execution of a report results in a time out.

You have a number of monitoring and supportability options available on the Windows Azure platform. These include the Windows Azure management console, Dashboard, which shows the CPU and Memory utilization of your website or Web Role. You can also configure and download IIS logs and Failed Request Tracing logs for offline analysis.

You also have the capability for Remote Desktop Connection to a Web Role/Cloud Service. By employing this, you can access the IIS Management console, which enables you to configure, monitor, and optimize the IIS configuration. The Remote Desktop Connection enables you to access tools, such as Task Manager, Event Viewer, and Performance Monitor to gain a deeper understanding of how your website performs.

WHAT YOU NEED TO USE THIS BOOK

To run the samples in the book, you need the following:

- Microsoft Visual Studio 2012 Express or Professional
- Microsoft SQL Server 2012
- NHibernate 3.3
- A Windows Azure Account

The source code for the samples is available for download from the Wrox website at `www.wrox.com/go/azureaspmvcmigration`

CONVENTIONS

To help you get the most from the text and keep track of what's happening, a number of conventions are used throughout the book.

www.it-ebooks.info

WARNING *Warnings hold important, not-to-be-forgotten information directly relevant to the surrounding text.*

NOTE *Notes indicate notes, tips, hints, tricks, or asides to the current discussion.*

As for styles in the text:

- We *italicize* new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show filenames, URLs, and code within the text like so: `persistence.properties`.
- We present code in two different ways:

```
We use a monofont type with no highlighting for most code examples.

We use bold to emphasize code that is particularly important in the present context or
to show changes from a previous code snippet.
```

SOURCE CODE

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All the source code used in this book is available for download at `www.wrox.com`. Specifically for this book, the code download is on the Download Code tab at `www.wrox.com/go/azureaspmvcmigration`.

You can also search for the book at `www.wrox.com` by ISBN (the ISBN for this book is **978-1-118-67858-9** to find the code. And a complete list of code downloads for all current Wrox books is available at `www.wrox.com/dynamic/books/download.aspx`.

At the beginning of each chapter, you can find a list of the major code files for the chapter.

Most of the code on `www.wrox.com` is compressed in a ZIP, RAR archive, or similar archive format appropriate to the platform. After you download the code, just decompress it with an appropriate compression tool.

NOTE *Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is* ***978-1-118-67858-9****.*

Alternatively, you can go to the main Wrox code download page at `www.wrox.com/dynamic/books/download.aspx` to see the code available for this book and all other Wrox books.

www.it-ebooks.info

ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you may save another reader hours of frustration, and, at the same time, you can help us provide even higher-quality information.

To find the errata page for this book, go to `www.wrox.com/go/azureaspmvcmigration` and click the Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors.

If you don't spot "your" error on the Book Errata page, go to `www.wrox.com/contact/techsupport.shtml` and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

P2P.WROX.COM

For author and peer discussion, join the P2P forums at `http://p2p.wrox.com`. The forums are a web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At `http://p2p.wrox.com`, you can find a number of different forums to help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to `http://p2p.wrox.com` and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join, as well as any optional information you want to provide, and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

NOTE *You can read messages in the forums without joining P2P, but to post your own messages, you must join.*

---

www.it-ebooks.info

After you join, you can post new messages and respond to messages other users post. You can read messages at any time on the web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to This Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works, as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.

www.it-ebooks.info

PART I
Migration

- **CHAPTER 1:** Migrating from ASP.NET to ASP.NET MVC 4
- **CHAPTER 2:** Creating the ASP.NET MVC 4 Project

www.it-ebooks.info

1 Migrating from ASP.NET to ASP.NET MVC 4

CONCEPTS

IN THIS CHAPTER

- Knowing what you need to start
- Understanding the difference between ASP.NET and ASP.NET MVC
- Seeing the differences between ADO.NET and NHibernate
- Exploring the improvements from IIS 7.0/7.5/8 to IIS 6
- Understanding Team Foundation Server
- Using Test Driven Development in ASP.NET MVC

Deciding to take the leap from one technology to another is a difficult one. You face the risk of improper implementation of the technology, which can cause performance issues. You also face possible impact to business logic, which can produce disruption to your customers and partners. Many core business processes were written many years ago and live on *legacy* environments; although the developers who wrote them are long gone, and the support teams who run the operations likely are offshore and have a high resource turnover rate. Over time, the cost of maintaining these legacy systems will increase, and the knowledge of how they work will continue to dwindle.

Although change from legacy systems to more robust platforms and newer technologies is difficult, the rewards in the form of cheaper operations cost and increased customer loyalty, as well as avoidance of risks, likely setbacks, and wrong turns are worth the effort. Not only that, but your employees will stay around longer and your developers will be happier because they get exposure to and are working on the newest technologies.

After you decided to migrate for all those good reasons, you can move to the following section, which discusses some of the differences between ASP.NET, ASP.NET MVC, ADO.NET, and NHibernate.

NOTE *This chapter contains key migration concepts. To learn how to create an actual project, see Chapter 2.*

---

www.it-ebooks.info

GETTING STARTED

When I first started migrating the sample ASP.NET blog website I planned on maintaining the same look and feel in the new ASP.NET MVC 4 website. A few web pages into the migration I started thinking, "Why am I changing technologies when, in the end, I will have the same website?" Most visitors to a website do not know what technology runs it. Customers or visitors know only if the website looks and performs well. So I decided to not only change the technology that produces my website, but also to give it a new look and feel. Improvements to your website keep your customers interested and coming back. So when you change, make sure the customer can easily notice the innovation.

This chapter discusses some important migration topics, which you will see in action in Chapter 2, such as:

- ASP/COM versus ASP.NET versus ASP.NET MVC
- ADO.NET versus NHibernate (Object Relational Mapping)
- IIS 6 versus IIS 8
- Team Foundation Server
- Test Driven Development in an ASP.NET MVC 4 application

NOTE *You should have a good understanding of ASP.NET, ADO.NET, and IIS. It should be clear to you that ASP.NET web forms dynamically present content based on user input, that ADO.NET is a framework for data retrieval and manipulation, and lastly, that IIS is the web server, which answers HTTP requests. An in-depth understanding of these technologies and their dependencies is expected. A deep comprehension of NHibernate, TFS, or development approaches is not as critical.*

COMPARING ASP.NET TO ASP.NET MVC

The choice to move away from ASP.NET web forms to a new technology, such as ASP.NET MVC 4, is not an easy one, and you may not even have the opportunity to make it; many companies have invested a lot into their ASP.NET web forms platform, enduring complexities and hard-fought challenges to get their system to where it is now, and they will likely meet the prospect of change with resistance. At the same time, some of these ASP.NET web form systems are simply old; some may even be ASP/COM, with the costs for supporting them likely to increase in the near future. Therefore, some actions in the short term must be made. While making such decisions and having discussions with managers and development teams, consider the concepts, technologies, and tools discussed in this section and chapter.

www.it-ebooks.info

A starting point is to discuss the advantages and disadvantages of both ASP.NET and ASP.NET MVC 4. The following four sections provide a brief overview. The following are the advantages of ASP.NET web forms over previously existing technology (ASP/COM):

- A lot of rich user controls that boost Rapid Application Development (RAD)
- Familiar model for Windows Forms developers
- Compatibility with any .NET language such as C# and VB.NET
- Support for object-oriented programming (OOP)
- Separation of presentation and business logic code
- Web Service capabilities

The best approach for understanding the advantages of a new technology is to analyze the technology the new one replaces. ASP.NET replaced or maybe, enhanced in many aspects, the ASP/COM platforms. If you recall, ASP/COM exposed only a number of built-in objects such as `Request`, `Response`, `Application`, and `Session`. These objects still exist and are useful in even the most sophisticated of systems. However, they had some problems, such as using the `Session` object in a web farm, some challenges with the `Response Buffering` implementations, and the ominous regsrv32.

Although some features found in these technologies were non-optimal, they were ground-breaking at the time. The release of ASP.NET resolved many of these sub optimal features. One such feature enabled the separation of your presentation layer from your business logic layer using the *code-behind concept*. A reference to the code-behind is made in the ASP.NET file (*.aspx), as shown in the following snippet, which tells the compiler where to reference the method and properties.

```
<%@ Page Title="As keyword in C#" Language="C#"
    CodeFile="As-keyword.aspx.cs" … %>
```

The snippet identifies a number of things, but the important attribute is the language of the code-behind and the name of the file, that is, *.aspx.cs or *.aspx.vb.

Overtime, as ASP.NET was implemented into highly scaled and highly trafficked websites, it became apparent that the state-full nature as well as the heaviness of the ASP.NET user controls and `VIEWSTATE` needed some additional attention and optimization. Some of these optimization techniques are discussed in Chapter 3, and some examples are implemented in Chapter 4.

In addition, the migration from ASP/COM to ASP.NET (which generally was created by those who wrote the more process-oriented ASP/COM code) was not designed from a model/object-oriented perspective. At that time, many C++ coders with strong OOP skills still focused mainly on client-server or Windows services development. So they delivered more process-oriented code on the ASP.NET platform.

www.it-ebooks.info

Potential disadvantages of ASP.NET web forms include:

- Support for a single `<form>` tag.
- By default, the `VIEWSTATE` is enabled and web pages can become bloated.
- Difficult source code manageability of very large websites and teams.

There may be more, but the point of this discussion is not to find fault with ASP.NET; most developers would agree that, based on the alternative, ASP.NET is a useful and feature-rich technology.

It's not correct to say that ASP.NET MVC is a replacement for ASP.NET. Rather, it is an alternative. You can fuse together ASP.NET MVC and ASP.NET to create a hybrid web-based application. Many enhancements are still being made to the ASP.NET features. An example of this is Web Tools 2012.2 and the .NET Framework, which is the platform on which you can create code customizations and innovations.

This book, and specifically this chapter, focuses on migrating a blog website using ASP.NET to an ASP.NET MVC 4 project, which is ultimately deployed as a Windows Azure Web Role/Cloud Service. Figure 1-1 shows the available features on the current ASP.NET website.

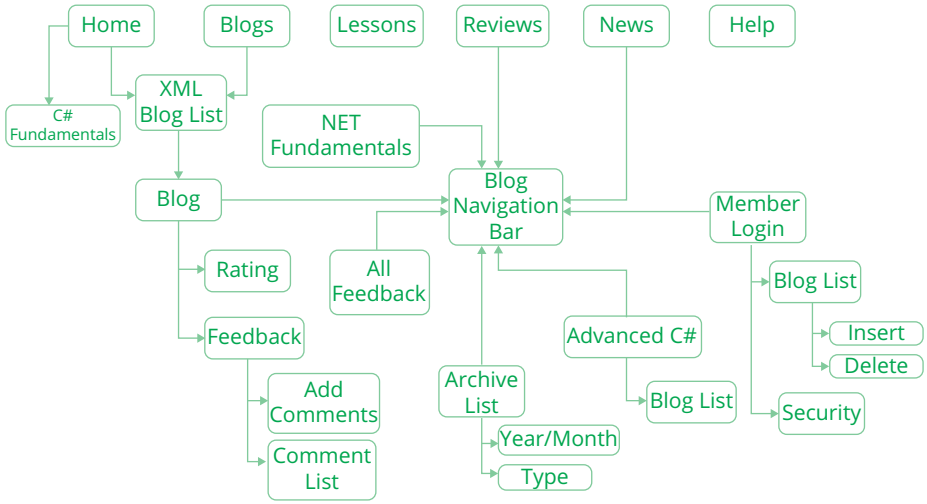


FIGURE 1-1

Although Figure 1-1 shows a relatively simple blog website, you can learn many of the important and challenging features in MVC by migrating this site to MVC. Chapter 2 provides many examples and exercises required to complete this migration, so you can get a better understanding of MVC concepts.

www.it-ebooks.info

Understanding the Model–View–Controller

A basic, but fundamental, concept in ASP.NET MVC is the MVC part. *MVC* stands for *Model–View–Controller* and is a pattern that helps designers and developers separate the system between input logic, business logic, and user interface (UI) logic. Figure 1-2 illustrates the MVC pattern, which consists of:

- **Model:** This is the class used to store, manipulate, and retrieve information from a database.
- **View:** Typically created from the model data, this is the application's user interface, displays the data, and supports the manipulation of it.
- **Controller:** Selects the view to display the data existing in the model and handles the interaction with the application user. Controls the interaction between the model and the view.

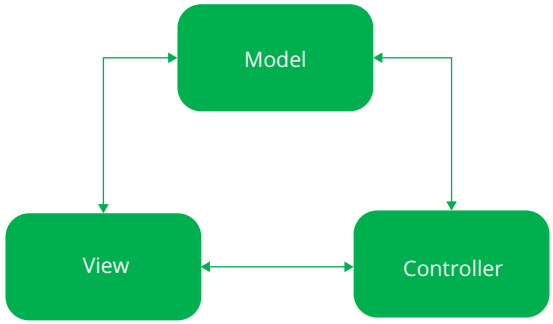


FIGURE 1-2

Having a global template for the website greatly reduces the amount of effort when you implement changes to the look and feel of your website. ASP.NET has *Master Pages*, which you create with the `<%@ Master` tag at the beginning of the page. You then place a `<asp:ContentPlaceHolder` tag in the master page where you want to display content. Although Master Pages are supported in an ASP.NET MVC system, you may want to consider using a layout page.

Using Master Pages versus Shared Views

When you create an ASP.NET MVC 4 project (a sample is shown in Figure 1-3), a view called _Layout.cshtml is generated in the View\Shared directory. The _Layout.cshtml file contains the

www.it-ebooks.info

<head>, <body>, <header>, and <footer>. Nonetheless, the Layout property of the System.Web.WebPages.StartPage class is set to _Layout.cshtml in the Views_ViewStart.cshtml, and is loaded with each rendering of a page. This is similar behavior of the ASP.NET master page.

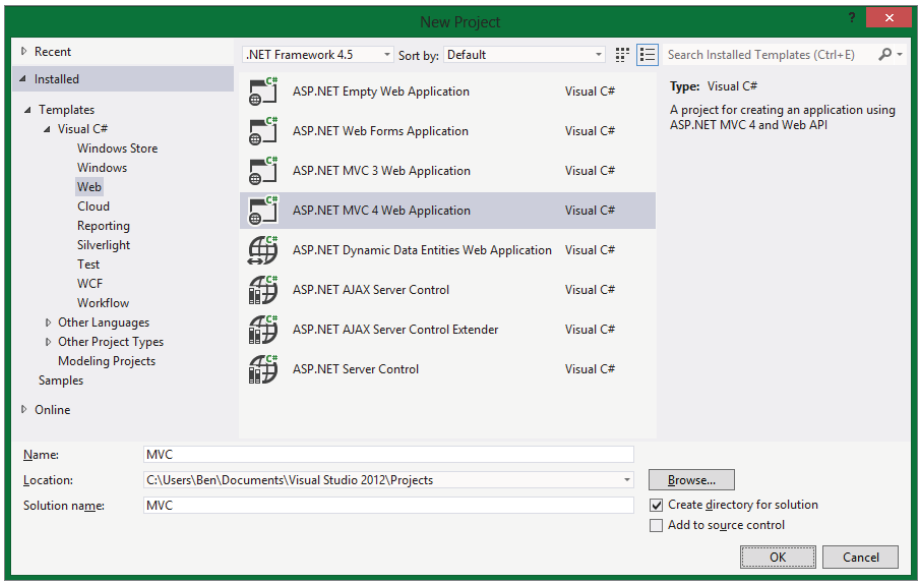


FIGURE 1-3

NOTE *If you haven't used HTML in some time, you might find a short refresher will be of benefit. For a refresher, you might consider reading* Beginning HTML and CSS, *(Wiley, 2013).*

*Also, take a look at the Razor View Engine and learn what it is and how to use it in ASP.NET MVC applications. The *.cshtml file extensions mean C# Razor.*

When you make a request, using the following code snippet for example, the content loads along with the Index.cshtml view located in the Views\CSharpFundamentals directory and the Index method in the Controller\CSharpFundamentalsController.cs file is called.

```
@Html.ActionLink("C# Fundamentals", "Index", "CSharpFundamentals")
```


www.it-ebooks.info

In Chapter 2, you'll find that the `@RenderBody()` in the _Layout.cshtml file is responsible for rendering the content in the Views\CSharpFundamentals\Index.html file.

When to Use User Controls versus Partial Views

Other useful features in ASP.NET are *user controls* (that is, *.ascx files), which enable you to group capabilities or content together and reference them from other files. For example, in the ASP.NET and ASP.NET MVC 4 code for this book, you'll find a Blog Navigation Bar that contains a translator, comment information, blog information, tag cloud, and blog archive. This Blog Navigation Bar is referenced from all the blogs and also in some of the other pages on the sample blog website. Instead of adding the code for this Blog Navigation Bar to each page, the page is referenced from each ASP.NET page using a code snippet similar to the following:

```
<%@ Register TagPrefix="NAVBAR" TagName="BlogBar"
    Src="~/include/BlogRightColumn.ascx" %>
…
<NAVBAR:BlogBar ID="NAV1" runat="server" />
```

Again, user controls are supported in an ASP.NET MVC 4 application. An alternative, and the one used for an exercise in Chapter 2, is the `@Html.Partial()` method. The view, which is a parameter of the `@Html.Partial()` method, is used like a control in the ASP.NET MVC 4 application and is created in the View\Shared directory. The file is called BlogNarBar.cshtml and contains all the code and content required to render the same information as rendered in the ASP.NET user control. The following code snippet demonstrates how to imbed the partial view into another page:

```
@Html.Partial("_BlogNavBar", ViewData["blognavbar"])
```

There are a number of different overloads for the `Html.Partial()` method. The one used here is the `Partial(HtmlHelper, String)`, where `HtmlHelp` is the name of the view to be rendered and the `String` is a dictionary called `ViewData`, which contains data and passes that data between a controller and a view.

Understanding the Statelessness of ASP.NET MVC

`VIEWSTATE` is a useful but often overused feature in ASP.NET, which can cause some serious performance issues. `VIEWSTATE` saves the state of the data in controls across page post backs. For example, you can store a drop-down list on the ASP.NET page that is populated by a database query in the `VIEWSTATE` and pass it between pages without having to re-query the database for its content.

`VIEWSTATE` is not supported in ASP.NET MVC. The reason for this might be for the reason just stated — that it's heavy and often overused. In addition, there's a move towards a more stateless architecture, enhanced integration with testing, and better alignment with the *separation of concerns (SoC)* concept.

www.it-ebooks.info

Understanding Strongly Typed Class References

Another nice feature of ASP.NET MVC 4 is strongly bound type references between the view and the model. The Razor View Engine provides the capability for displaying the data in the model using a strong reference. The Razor View Engine was introduced in MVC 3 to simplify the presentation of data in the view. For example, Listing 1-1 shows the `@model List<MVC.Models.BlogList>` collection being referenced in a .cshtml file using a `foreach` statement.

LISTING 1-1: Using Razor to Display a Strongly Typed Class

```
@foreach (var item in Model)
{
  <tr><th>@item.Title</th></tr>
  <tr><td>@item.Description</td></tr>
  <tr><td><a href="@item.Url">More...>>></a></td></tr>
}
```

When you add the items to the `foreach` statement, IntelliSense is also supported.

NOTE *For more information about the Razor View Engine consider purchasing a book focused specifically on ASP.NET MVC 3 or 4. A good start is* Professional ASP.NET MVC 4 *(Wiley, 2012).*

ASP.NET, ASP.NET MVC, or Both?

Some additional advantages of ASP.NET MVC 4 for further consideration are:

- The separation of content from presentation (SoC)
- Better support for Test Driven Development
- URL's optimized for SEO and REST integrated with MapRoutes
- No server-based forms
- Good ALM and TFS integration for use with large development teams

The decision whether to use ASP.NET or ASP.NET MVC is a difficult one. You should review the previous section and utilize other resources (such as consultants or online forums for posting any question you have) to make the decision best for your situation. It is common to have hybrid systems that use both ASP.NET web forms together with ASP.NET MVC, Model–View–Controller capabilities. Taking the best parts of each and building a most optimal solution may be the way to go for you.

www.it-ebooks.info

WHAT ARE THE DIFFERENCES BETWEEN ADO.NET AND NHIBERNATE?

You'll encounter a number of differences between an ADO.NET data access layer and Object Relational Mapping (ORM)–like NHibernate. The main difference is the focus on design first versus model first. Before the creation or existance of ORMs software, development was usually driven by a relational-database. This caused some problems when the ASP.NET developer attempted to map the code to the database which could and often did change. The ORM concept approaches the design of software development from a code first, object-oriented perspective. All changes made to the design of a database happen from the source code and you execute commands on data table structures.

Therefore, it is recommended that where a system already has an existing relational-database that you not attempt to rewrite the system using the model-first capabilites. Instead, either use an alternative library for accessing the data, such as ADO.NET or use the manual mapping methods available in either the NHibernate or the ADO.NET Entitiy Framework.

NOTE *For deeper analysis and review of NHibernate and the differences it has with ADO.NET you might consider reading* Working with NHibernate 3.0 *(Wiley, 2011).*

From a code perspective, the following sections discuss how you'd implement an ADO.NET data access layer versus a similar situation for an NHibernate ORM structure. These sections compare and contrast these concepts to give you a better understanding of the differences.

Understanding the Data Access Layer

A significant migration opportunity has to do with the data access layer. A *data access layer* is typically a group of methods that manage the access to the database, the execution of SQL queries, and the storage of that data used for rendering to the client. Objects such as result sets, connection, and command objects should sound familiar as should the following code snippet:

```
set conn=Server.CreateObject("ADODB.Connection")
```

In the programming world, there's been a lot of progress since this set of ADODB COM objects was first created. For example, its successor, ADO.NET, is a feature-rich set of classes available in the .NET Framework, which you access using the `System.Data` directive.

The data access layer provided in the sample ASP.NET website uses a common data access pattern. All the code making the connection executes queries and returns the results, storing them in the App_Code\DataAccess directory. This directory has a class called `ConnectionManager` that is used for all the methods, which make connections and execute queries against the database. Listing 1-2 is an example of an ADO.NET `ConnectionManager`.

www.it-ebooks.info

LISTING 1-2: ADO.NET ConnectionManager

```
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

internal sealed class ConnectionManager
{
  public static SqlConnection GetConnection()
  {
    // Build the connection string from Web.Config file
    try
    {
       string connectionString = ConfigurationManager
              .ConnectionStrings["tbcspitw-pro"].ConnectionString;

       SqlConnection connection = new SqlConnection(connectionString);
       connection.Open();
       return connection;
    catch(ConfigurationErrorsException)
    {
       //log the exception
    }
  }
}
```

Notice that the `ConnectionString` is stored in the web.config file. You can pass the connection string as a string directly into the `SqlConnection` object; however, if you do this and you need to change the password or server name, a code change and code migration are required. Saving the connection string in the web.config file enables you to make changes like this without the need to modifiy, test, and deploy new source code.

Using the `GetConnection()` method in the `ConnectionManager` class is typically done in a `using{}` statement, as shown in Listing 1-3.

LISTING 1-3: Using the GetConnection() Method of the ConnectionManager Class

```
using System.Data.SqlClient;
using System.Data;

public static int PostTotalCount()
{
  int count = 0;
  string sql =
        "SELECT COUNT(DISTINCT POSTID) FROM POSTS WHERE TYPE ='blogs'";

  using (SqlCommand command =
```


www.it-ebooks.info

LISTING 1-3: ***(Continued)***

```
        new SqlCommand(sql, ConnectionManager.GetConnection()))
  {
    command.CommandType = CommandType.Text;
    count = (int)command.ExecuteScalar();
  }
  return count;
}
```

This is a relatively common ADO.NET data access layer implementation. Since the ADO.NET data access layer concepts were first implemented, the introduction of LINQ, repositories, and data models have greatly enhanced this piece of functionaity. ADO.NET is still a very valid and proven data access solution. The driving factor for change to an ORM is a much tighter coupling between the code and the database. Notice that in Listing 1-3 the SQL Query is hard-coded and string-based. This is prone to problems when columns and or tables change on the database. For that reason, in ADO.NET the binding between the code and the database is much weaker.

Understanding the Object Relational Mapping

An *Object Relational Mapping (ORM)* is an object-oriented approach for designing, creating, and retrieving data from a data source. Two of the most popular ORM's are NHibernate and the ADO.NET Entity Framework. Some advantages of an ORM follow:

- Development is faster because ORM reduces repetative code resulting in reduced costs and time.
- It's not dependent on a single database vendor (excluding EF).
- When properly implemented, ORMs have solid caching and pooling capabilities.

Some disadvantages of an ORM follow:

- The learning curve can be steep.
- It's challenging to write complex SQL queries.
- It can be a little slower than non-ORM solutions.

NOTE *This chapter and Chapter 2 focus on the transition from ADO.NET to NHibernate. However, the ADO.NET Entity Framework, which is Microsoft's equiviliant to NHibernate. Both do similiar things. Therefore, you should look at both to determine which one best meets your needs.*

---

www.it-ebooks.info

Regarding the learning curve, a disadvantage of NHibernate, when compared to ADO.NET Entity Framework, is the setup complexity. However, after it is performed a few times, the process is relatively straightforward, whereas the ADO.NET Entity Framework is ready to go out-of-the-box.

The setup of NHibernate in a web-based scenerio is generally done in the Global.asax.cs file. An example of the configuration is shown in Listing 1-4. You do need to install the NHibernate binaries prior to using this code.

LISTING 1-4: NHibernate Setup Configuration Example

```
protected void Application_Start()
{
  …
  var configuration = ConfigureNHibernate();
  configuration.CurrentSessionContext<WebSessionContext>();
  HbmMapping mapping = GetMappings();
  configuration.AddDeserializedMapping(mapping, "ASP.NET.MVC.4");
  SchemaMetadataUpdater.QuoteTableAndColumns(configuration);
  SessionFactory = configuration.BuildSessionFactory();
  try
  {
    new SchemaExport(configuration).Drop(false, true);
    new SchemaExport(configuration).Create(false, true);
    SchemaValidator schemaValidator =
                    new SchemaValidator(configuration);
    schemaValidator.Validate();
  }
  catch (HibernateException e)
  {
    //Log the error in the Event Viewer using
    //the System Diagnostics library
  }
  …
}
```

In Chapter 2, you are guided through the configuration and implementation of NHibernate. Therefore, all the methods are not displayed here. Even so, it is important to call attention to a few methods, for example the `CurrentSessionContext` type. In the sample solution the `WebSessionContext` context is chosen, but you need to understand the following as well:

- `NHibernate.Context.ManagedWebSessionContext`: This session is tracked by the `HttpContext` object. You are responsible for binding and unbinding the `Session` methods to the class.
- `NHibernate.Context.CallSessionContext`: The session is tracked by the `CallContext`. You are responsible for binding and unbinding the `Session` methods to the static methods of the `CurrentSessionContext` class.

www.it-ebooks.info

- `NHibernate.Context.ThreadStaticSessionContext`: The session is stored as a static thread variable and supports a single session factory. You are responsible for binding and unbinding the `Session` methods to the static methods of the `CurrentSessionContext` class.
- `NHibernate.Context.WebSessionContext`: This is the same as `ManagedWebSessionContext`.

The context type is an important factor to investigate further when you plan on implementing any type of level 1 or level 2 caching. The chosen implementation for this sample creates a `Session` per database transaction, and there is no caching or lazy loading happening with this approach. If you need either of these capabilities in your ASP.NET MVC 4 web solution, choose another context and configure it appropriately.

NOTE *In Listing 1-4, it is important that you comment out the* `SchemaExport.Drop()` *and* `Create()` *methods after the application is run once. As coded, each time the application is started, the database will be dropped and re-created. You will lose all data.*

Understanding Mapping by Code

Another "newer" feature in NHibernate 3.0 is *mapping by code*. The approach originally required (and still supports) mapping files based on XML, which provide information about each column type and the relationship between other entities. These XML mapping files are no longer required; instead, you can perform mapping in a C# class file. Listing 1-5 provides an illustration of an NHibernate entity mapped by code.

LISTING 1-5: NHibernate — Mapping by Code

```
using NHibernate.Mapping.ByCode;
using NHibernate.Mapping.ByCode.Conformist;
namespace MVC.Models
{
  public class Blog
  {
    public Blog()
    {
       comments = new List<Comments>();
    }

    public virtual int Id { get; set; }
    public virtual string Title { get; set; }
```


www.it-ebooks.info

LISTING 1-5: *(Continued)*

```
    public virtual string Type { get; set; }
    …
    public virtual IList<Comments> comments { get; set; }

  }

  public class BlogMap : ClassMapping<Blog>
  {
   public BlogMap()
   {
      Id<int>(x => x.Id);
      Property<string>(x => x.Title);
      Property<string>(x => x.Type);
      …

      Bag<Comments>(x => x.comments, cp => {},
                    cr => cr.OneToMany(x => x.Class(typeof(Comments))));
    }
  }
}
```

Notice that the columns and their types are identified in the class. In addition, the relationships between the `Blog` class and the `Comments` class are created in the mapping class. When the `SchemaExport.Create()` method is run, the columns and primary and foreign keys are created.

Accessing Data from the Database

After the configuration of NHibernate is complete and the associated database schemas exist on the database, the next phase in a migration from an ADO.NET data access layer to NHibernate is to retrieve data from the database. NHibernate has a LINQ data access provider, referred to as LINQ to NHibernate. There are a number of different LINQ providers:

- LINQ to NHibernate
- LINQ ro Entities (ADO.NET Entity Framework)
- LINQ to SQL
- LINQ to XML

NHibernate has other interfaces such as `IQuery` and `ICriteria` that give you different capabilites and options for accessing your data. An example of a LINQ to NHibernate query is shown in Listing 1-6.

www.it-ebooks.info

LISTING 1-6: LINQ to NHibernate Example

```
public IList<Blog> GetAdvancedBlogs()
{
  using (ISession session = MvcApplication.SessionFactory.OpenSession())
  using (ITransaction transaction = session.BeginTransaction())
  {
    IList<Blog> blogs = (from b in session.Query<Blog>()
                         where b.Advanced == 1
                         select b).ToList<Blog>();

    return blogs;
  }
}
```

Choosing to implement an ORM is not only a change in technology but also a change in the approach used to create and design your application. In many cases the database design and the relationships between the tables dictate or drive the capabilities of an application. However, using an ORM, the tables (☺) are turned, and the objected-oriented design of your application drives the database and its relationships.

EXAMINING INNOVATIONS FROM IIS 6 TO IIS 7.0/7.5/8

Internet information Service (IIS) has been around for many years and during that time has matured greatly. One of its only significant competitors is Apache, which usually runs on a Linux machine. IIS acts as a web server that responds to HTTP requests using verbs such as `GET`, `POST`, `HEAD`, and `PUT`.

If you review IIS versions 6 and IIS 7+, you'll find a lot of differences and particularly new capabilities and enhancements. This section summarizes two of the most important enhancements from a migration perspective.

NOTE *For more information on the various IIS 7+ capabilities, consider doing an Internet search or purchasing a book specifically about IIS. A good start is* Professional IIS 7, *(Wiley, 2008) or the* Professional IIS 8, *(Wiley, 2012).*

Before discussing the differences between the IIS versions, you should know that when you create a Windows Azure Cloud Service, you can choose from a number of different Windows

www.it-ebooks.info

Server versions, as shown in Figure 1-4. The choices and the associated IIS versions as displayed as follows:

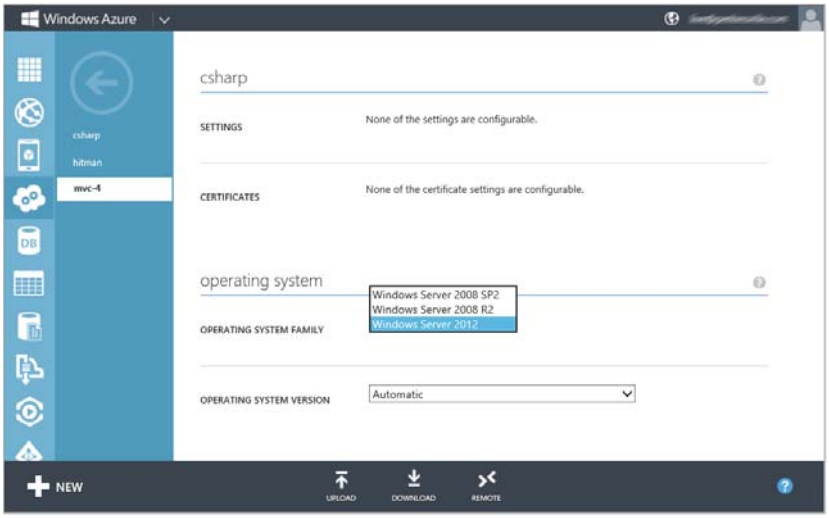


FIGURE 1-4

- **Windows Server 2008 SP2:** IIS 7
- **Windows Server 2008 R2:** IIS 7.5
- **Windows Server 2012:** IIS 8

One significant difference between IIS 6 and IIS 7+ is the processing of these HTTP requests through what is known as the request pipeline. The *request pipeline* is a set of ordered native and managed modules that review or react to an HTTP request running through it. Some native modules are:

- AnonymousAuthenticationModule
- WindowsAuthenticationModule
- HTTPCacheModule
- TracingModule

Some examples of managed modules are:

- FileAuthorization
- Session
- UrlMappingsModule

The type of module you use (that is, native versus managed) is important because it's version-specific due to the integrated pipeline. For example, in IIS 6 when you request a managed

www.it-ebooks.info

resource, the authentication process happens in both the ASP.NET pipeline and the IIS pipeline, which is not optimal. In IIS 7+, the appropriate authentication modules (managed or native) are used and authentication is performed only once.

IIS 7+ still supports processing of a request as it happens in IIS 6. Some systems have been developed and are dependent on this mode and would not work in IIS 7+. To have your application run in IIS 6 mode, set your application pool to run in Classic mode, as shown in Figure 1-5.

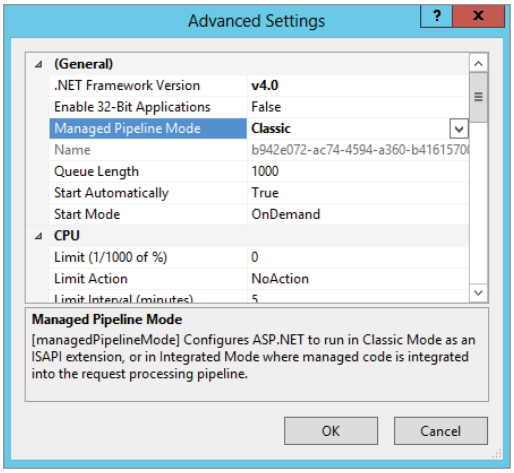


FIGURE 1-5

To have your system run in the new unified pipeline mode, leave the application running in Integrated mode, which is the default. The second important difference is the change in built-in user and groups accounts. Table 1-1 describes the IIS 6 and IIS 7+ user group equivalents.

TABLE 1-1: Built-in IIS User and Group Accounts

IIS 6	IIS 7+
IUSR_MachineName	IUSR
IIS_WPG	IIS_IUSRS

The IIS 6 account and group are both local accounts, and when created, are associated to a *security identifies (SID)* — a unique number that identifies it from other accounts. A problem with IIS 6 is that this SID is stored in the metabase.xml file, and when you attempt to copy your configuration between one server and another, it does not work because the account and group on the new server has a different SID.

You can work around this by creating a domain account under which your application pool is run, and add this to the IIS_WPG groups. However, this means you need to add an Active Directory (AD)

www.it-ebooks.info

to your architecture. In cases in which a web farm or web server exists in a neutral zone or DMZ, this creates a significant overhead, which increases support and maintenance efforts.

The IUSR is an anonymous account that works similar to the LOCAL SERVICE account and that has no access to resources on the network. This account is used to execute requests running under the Anonymous Authentication provider. Every request needs to run under an identity.

The IIS_IUSRS group in IIS 7+ will have the same SID on Windows Server 2008. Therefore, when you attempt to copy your configuration from one server to the next, you should have no problems with the ACL on the file and system resources.

As mentioned, there are numerous differences between IIS 6 and IIS 7+. When making the jump, you need to choose the newest version of IIS because it has the most current capabilities and features. In addition, you should purchase an IIS book that covers these differences in more details.

INTRODUCING TEAM FOUNDATION SERVER

For many developers, the memories of Visual Source Safe (VSS) are burned into our brains like the theme song from your favorite television or radio episode. In the early days of software development activities, there were not so many source code repositories and VSS performed a key role, not only by keeping a company's most valuable intellectual property safe, but also by being the basis of the entire release management process.

The Waterfall development process was one of the only methods in existence for moving code from thought to production. You could check code in and out, making a branch to fix bugs, releasing a new bug fix version on a monthly schedule, and releasing new features on a quarterly schedule. However, merging the code back together after releases was truly an art form, especially when there was overlap.

Today, you have many types of development processes, such as Agile, Extreme Programming, Continuous Integration, and so on. The list goes on but the goal is the same: To get the code checked in, tested on the different environments, such as system tests and integration tests, and then get code into production as fast as possible.

Visual Source Safe is no longer. It has been replaced by a very feature-rich and flexible solution called *Team Foundation Server (TFS)*. This tool provides much more than a source code repository for managing the version of your source code. It provides features such as the following:

- Grouping activities into Team Projects
- Building management and testing capabilities
- Using project management tools and work items
- Using project reporting and metrics
- Integrating with Windows Azure

www.it-ebooks.info

Team Foundation Server is coupled somewhat to *Application Lifecycle Management (ALM)*, which governs the life of an application from requirements gathering to maintenance. The concept of *Continuous Integration* is a focal point of both these solutions, and gets the code through the cycles and into production as fast as they are coded.

Setting up the link between Windows Azure and TFS is relatively straightforward. Simply follow these steps:

1. Access your Windows Azure account.
2. Select the Cloud Service where you want to publish your code.
3. Select the Set Up TFS Publishing link, as illustrated in Figure 1-6.
4. Follow the steps provided in the Setup Wizard.

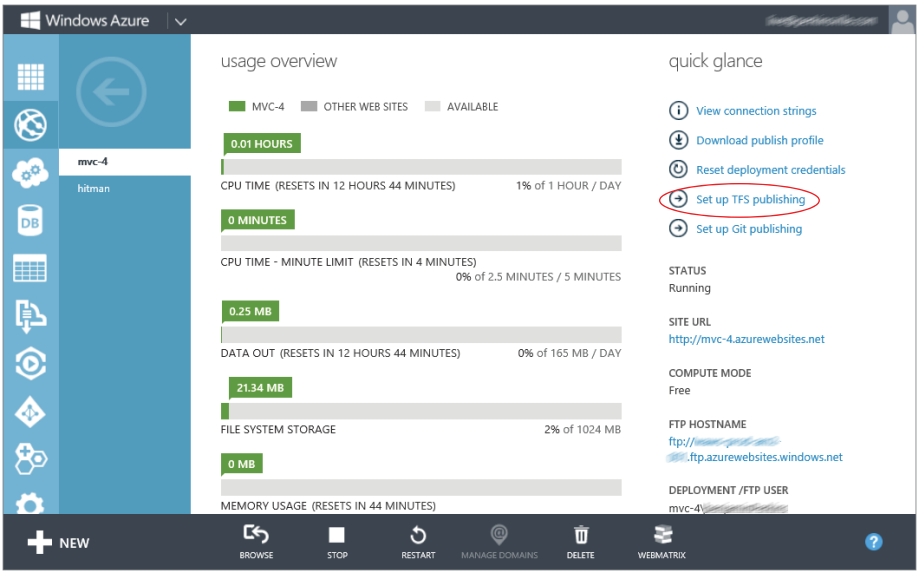


FIGURE 1-6

After you connect Team Foundation Server and the Windows Azure Web Site or Cloud Service together in Visual Studio 2012, you check out a piece of code, modify it, test it, and check it back in, and the application will automatically build and publish.

Figure 1-7 shows the source code Explorer view of a sample project that's being managed. The code, work items, builds, and testing can also be managed in the web component of TFS. For example, after you set up your TFS account, you can access your TFS at `http://uniqueName.visualstudio.com`. There you'll see the windows illustrated in Figure 1-8.

www.it-ebooks.info

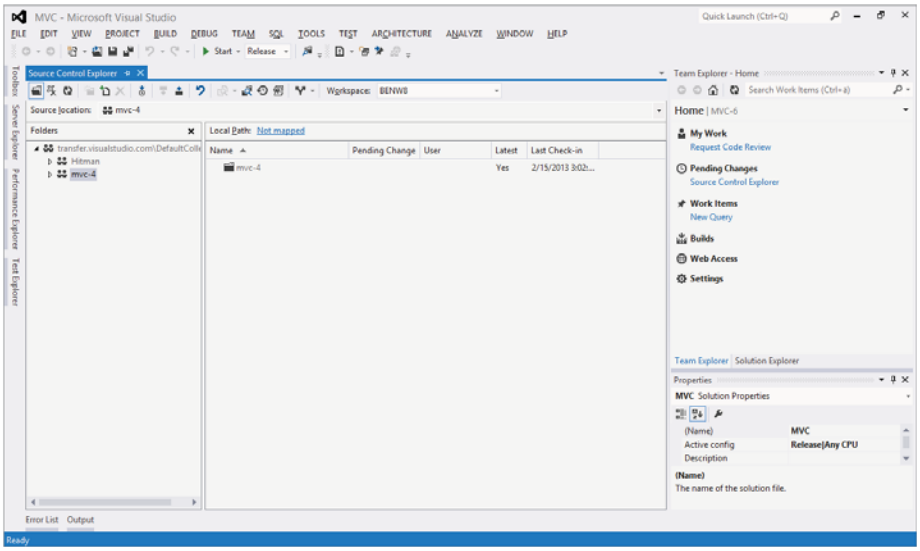


FIGURE 1-7

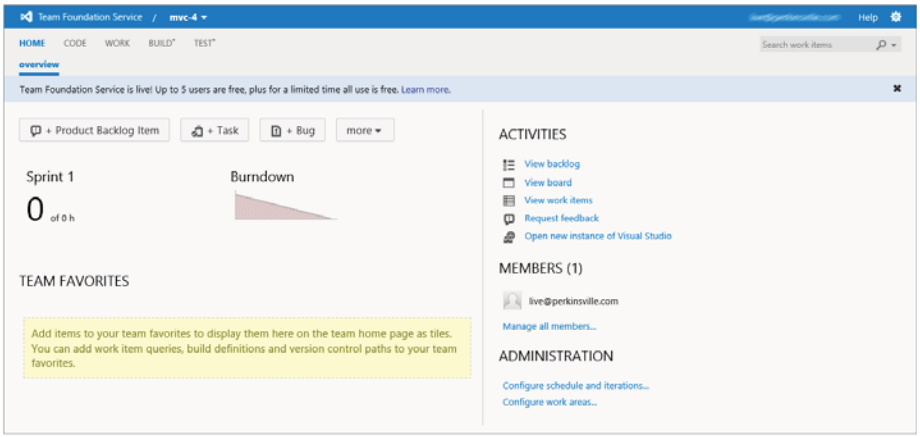

Team Foundation Service / mvc-4
Help
HOME CODE WORK BUILD TEST
Search work items
overview
Team Foundation Service is live! Up to 5 users are free, plus for a limited time all use is free. Learn more.
+ Product Backlog Item
+ Task
+ Bug
more
Sprint 1
0 of 0 h
Burndown
TEAM FAVORITES
Add items to your team favorites to display them here on the team home page as tiles. You can add work item queries, build definitions and version control paths to your team favorites.
ACTIVITIES
View backlog
View board
View work items
Request feedback
Open new instance of Visual Studio
MEMBERS (1)
live@perkinsville.com
Manage all members...
ADMINISTRATION
Configure schedule and iterations...
Configure work areas...

FIGURE 1-8

The Team Foundation Service has everything you need to manage your projects, report the status, and manage the source code and deployment onto the Windows Azure platform. There is no doubt that using these tools can result in getting the code into production faster than ever before.

www.it-ebooks.info

USING TEST DRIVEN DEVELOPMENT WITHIN ASP.NET MVC

ASP.NET MVC solutions are tightly connected to the *Test Driven Development (TDD)* model. Notice, as shown with Figure 1-9, that when you choose to create a new ASP.NET MVC Project, a generated unit test project is bound to the application by default.

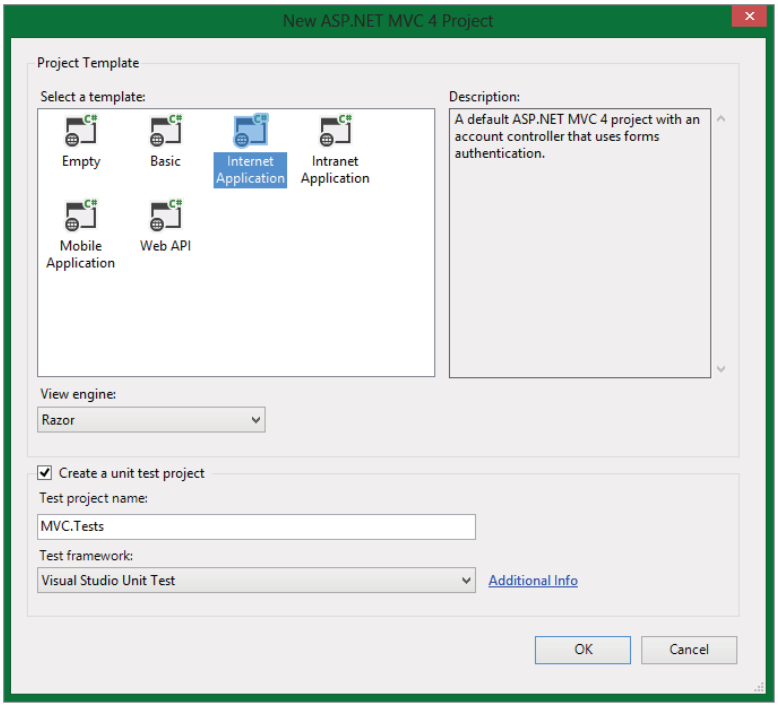


FIGURE 1-9

Test Driven Development is focused on writing the least amount of code necessary to meet the requirement. Common practice is to write the test case first and then write the code, which ultimately makes the test case a success. In many cases, development is done the other way around, that is, develop the code and then create the case to test it.

Listing 1-7 illustrates a test case that checks if there are any blogs written in December 2014.

www.it-ebooks.info

LISTING 1-7: Archive List for December 2014 — TDD

```
[TestMethod]
public void ArchiveList()
{
  BlogsController bController = new BlogsController();
  var result = bController.ArchiveList("2014", "DEC");
  Assert.IsNotNull(result);
}
```

After the test method code is implemented, you press CTRL+R, A to execute the test. You can expect the test to fail with the result shown in Figure 1-10.

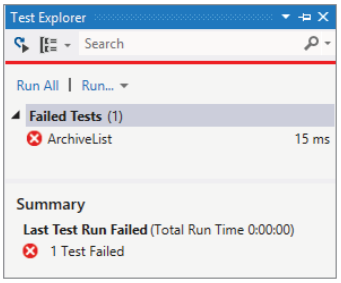


FIGURE 1-10

The process now is to create an `ArchiveList()` method that accepts a Year and a Month as parameters where blogs are known to exist. You then select the parameters from the data source. When the method is completed, you can re-execute the tests that is, CTRL with the expected outcome, as shown in Figure 1-11.

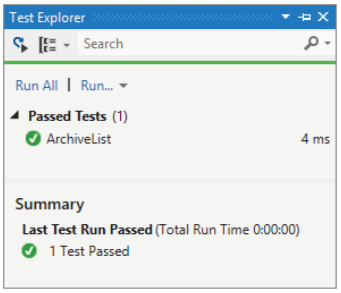


FIGURE 1-11

www.it-ebooks.info

SUMMARY

This chapter discussed the importance of innovation and change as well as the complexities and challenges you are likely to confront in the coming years due to retirement of Windows Server 2003, Windows XP, and IIS 6. Understanding the importance of the right platform for your application, as well as the opportunities posed by advancement in technologies, require your serious review and consideration. The recommendation of this chapter is that you not only change platforms (which brings more system stability, maintainability, and supportability), but also orient the culture and mentality of your development team to the new approaches and technologies.

Instead of simply moving your ASP.NET 2.0 application from IIS 6 to IIS 7, change where it makes sense! Use Windows Azure, use an ORM, use a development life cycle such as Agile or Continuous Integration, and use the tools that link all these things together, such as TFS and Visual Studio 2012. Innovate and create new value for your company and for your customers.

The next chapter has exercises that walk you through the migration from ASP.NET to ASP.NET MVC 4 and ADO.NET to NHibernate. After you complete this chapter's exercises, you should have a nice web application that you can then optimize for performance, deploy to the Windows Azure platform, and then monitor to ensure visitors have a pleasant visit.

www.it-ebooks.info

2 Creating the ASP.NET MVC 4 Project

EXERCISES AND EXAMPLES

IN THIS CHAPTER

- Modifying the _Layout.cshtml to get a nice "somewhat" unique look and feel
- Adding the methods required to list the blogs from the XML RSS file on the Home Index.cshtml page using Razor
- Adding the `Html.ActionLinks` (C# Fundamentals, .NET Fundamentals, Advanced C#, and so on), and corresponding models, views, and controllers
- Creating a local test SQL Server database called Blogs
- Implementing NHibernate into the ASP.NET MVC 4 application
- Creating the BlogNavBar partial view and add to a web page
- Creating the Archive List, custom MapRoute and ViewData containers
- Migrating a blog entry from ASP.NET with feedback form and comment list

WROX.COM CODE DOWNLOADS FOR THIS CHAPTER

You can find the Wrox.com code downloads for this chapter at `www.wrox.com/go/azureaspmvcmigration` on the Download Code tab. It is recommended that you download the code, review it so that you have a good understanding of what is about to be discussed. The steps in this chapter required to migrate the sample ASP.NET 2.0 website located at `http://aspnet.thebestcsharpprogrammerintheworld.com` to an ASP.NET MVC 4 application are provided in detail.

www.it-ebooks.info

The initial approach for this chapter was to make the ASP.NET MVC 4 application have the same look and feel as the ASP.NET 2.0 website. But after spending some hours doing this, I made the decision to use the default template because: for one, it is different, but mostly because the whole intent of this book and the concept around it is to innovate, change, and improve what already exists. I am not a fan of change for the sake of change, but where you can change and something good comes from it, I am all for it. Therefore, the new blog website will have the look and feel of the ASP.NET MVC 4 template.

NOTE *If you are not familiar with the concepts behind this chapter, please read through Chapter 1 before continuing.*

Also, it is highly recommended that you download the ASP.NET and ASP.NET MVC 4 source code and review the code as you simultaneously walk through these instructions. Every step required to make the transition from ASP.NET to ASP.NET MVC 4 are not included.

CHANGING THE LOOK AND FEEL OF YOUR WEBSITE

As you know, building a website to meet the look and feel of your product or your client can be time-consuming. Fortunately, ASP.NET MVC has default projects, so you can save time and headaches by starting with a default project and changing the look and feel to suit your needs. Plus, keeping your website current with changes in style and customer tastes makes it much more appealing and keeps customers and visitors returning.

For this section, you start with an APS.NET MVC 4 project as illustrated in Figure 2-1. You then make changes to the look and feel of the website. For example, you might changes the font size and type of colors. The changes you make are performed primarily to two files: the `_Layout.cshtml` located in the Views\Shared folder and the Site.css located in the Content folder. These two files contain a majority of the styling attributes used globally across the website.

At the same time, you would change the content of this page (Views\Shared_Layout.cshtml) to make it similar to what currently exists on the sample ASP.NET website.

NOTE *The MVC source code for this chapter contains the completed website. To begin this exercise, create an ASP.NET MVC 4 (Internet Application) project within Visual Studio.*

To make these changes, start at the top and work your way down by first opening the Views\Shared_Layout.cshtml file. After all the changes, have been implemented the website will resemble the page that appears in Figure 2-2. These changes are described in the following numbered steps:

1. Change the content after the `@ViewBag.Title` to be the title of the website.
2. Replace the default favicon.ico file in the root folder to the one for this website.

www.it-ebooks.info

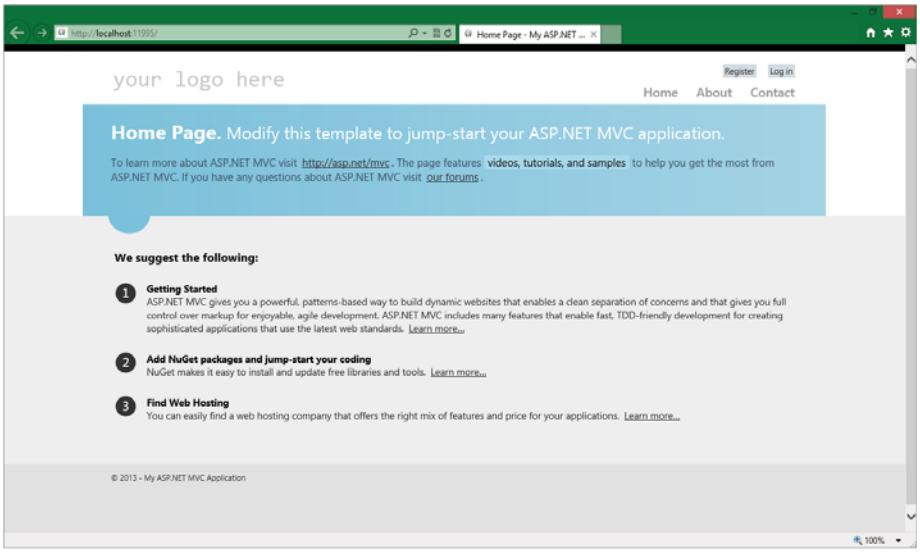

Home Page - My ASP.NET ...
your logo here
Register
Log in
Home
About
Contact
Home Page. Modify this template to jump-start your ASP.NET MVC application.
To learn more about ASP.NET MVC visit http://asp.net/mvc. The page features videos, tutorials, and samples to help you get the most from ASP.NET MVC. If you have any questions about ASP.NET MVC visit our forums.
We suggest the following:
1
Getting Started
ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives you full control over markup for enjoyable, agile development. ASP.NET MVC includes many features that enable fast, TDD-friendly development for creating sophisticated applications that use the latest web standards. Learn more...
2
Add NuGet packages and jump-start your coding
NuGet makes it easy to install and update free libraries and tools. Learn more...
3
Find Web Hosting
You can easily find a web hosting company that offers the right mix of features and price for your applications. Learn more...
© 2013 - My ASP.NET MVC Application
100%

FIGURE 2-1

3. Modify the Your Logo Here text to match the title of the website.
4. Open the Views\Shared_LoginPartial.cshtml and modify the existing linkText.
5. Open the Views\Home\Index.cshtml file and make the following modifications:
   - Change the `ViewBag.Title` to **C#**. You can find the `ViewBag.Message` in the Controllers\HomeController.cs file.
   - Modify the content of the page as required.
6. If necessary, change the color of the heroAccent.png file to match the color of the website.
7. Modify the contents of the footer.

Figure 2-2 shows the results of these steps which is the completed homepage.

You'll spend most of your time in the Content\Site.css file. It is recommended that you do not make the look and feel changes directly to a view. Instead make all the styling happen in the CSS file and then reference the CSS class from within the view.

For example, to change the default blue color to gray, you would change the class references in the Views\Home\Index.cshtml file. The referenced class is `.featured .content-wrapper`. Open the Content\Site.css file and search for what is illustrated in Listing 2-1. When the file is opened, enter CTRL+F and enter **`.content-wrapper`** to quickly find the section of code.

www.it-ebooks.info

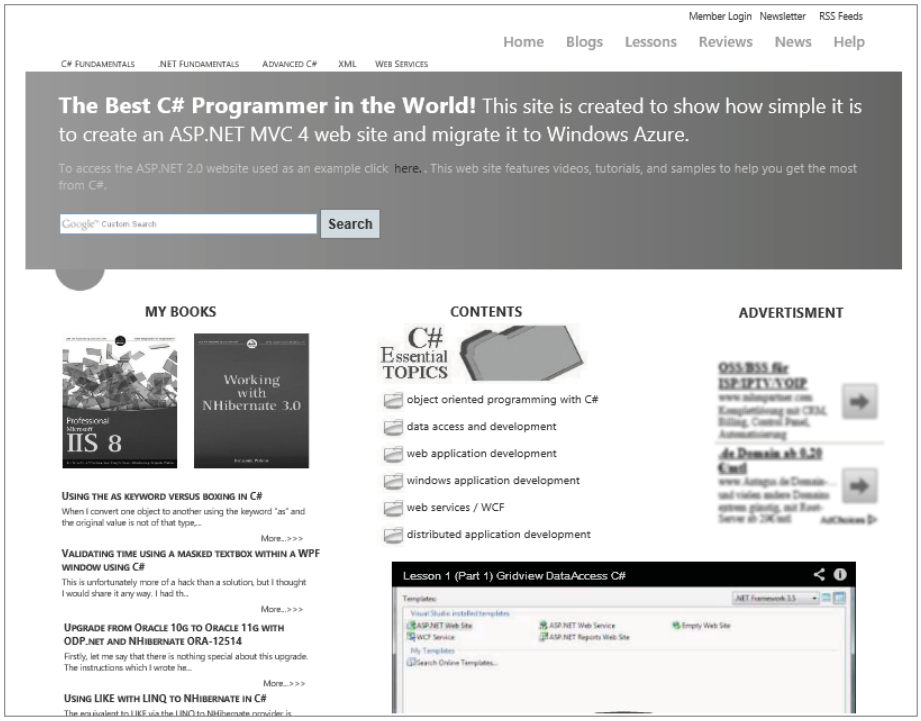

Member Login Newsletter RSS Feeds
Home Blogs Lessons Reviews News Help
C# FUNDAMENTALS .NET FUNDAMENTALS ADVANCED C# XML WEB SERVICES
The Best C# Programmer in the World! This site is created to show how simple it is to create an ASP.NET MVC 4 web site and migrate it to Windows Azure.
To access the ASP.NET 2.0 website used as an example click here. . This web site features videos, tutorials, and samples to help you get the most from C#.
Search
MY BOOKS
CONTENTS
ADVERTISMENT
C# Essential TOPICS
object oriented programming with C#
data access and development
web application development
windows application development
web services / WCF
distributed application development
Lesson 1 (Part 1) Gridview DataAccess C#

FIGURE 2-2

LISTING 2-1: Example of Site.css File Featured Class

```
/* page elements
-----------------------------------------------------------*/
/* featured */
.featured {
    background-color: #fff;
}

    .featured .content-wrapper {
        background-color: #7ac0da;
        background-image: -ms-linear-gradient(left, #7ac0da 0%, #a4d4e6 100%);
        background-image: -o-linear-gradient(left, #7ac0da 0%, #a4d4e6 100%);
        background-image: -webkit-gradient(linear, left top, right top,
                          color-stop(0, #7ac0da), color-stop(1, #a4d4e6));
        background-image: -webkit-linear-gradient(left, #7ac0da 0%, #a4d4e6 100%);
        background-image: linear-gradient(left, #7ac0da 0%, #a4d4e6 100%);
        color: #3e5667;
        padding: 20px 40px 30px 40px;
```


www.it-ebooks.info

LISTING 2-1: *(Continued)*

```
}

    .featured hgroup.title h1, .featured hgroup.title h2 {
        color: #fff;
    }

    .featured p {
        font-size: 1.1em;
    }
```

You would change the `#7ac0da` and `#a4d4e6` to the color you want as the theme for the website. In this example the website is black and white therefore you use `#000000` and `#FFFFFF` in all places where website colors are referenced.

NOTE *To preserve the transparency of your images, use a program such as Paint.NET, such as the heroAccent.png image. If you make these changes using Paint, the background will likely be white.*

The migration of the look and feel from the ASP.NET website and ASP.NET MVC 4 project are not covered in completed detail. Be sure that you are happy with the layout of the default website look and feel before continuing to the next section. The reason that this section does not give the steps for changing the look and feel in complete detail is so that you can set things up as you desire. Changing the outward appearance of a website does not require coding, it requires only HTML and CSS layout design and therefore the steps are not provided in detail. Instead, a sample converted website is provided for reference at `http://mvc-4.cloudapp.net`.

CREATING THE BLOG LIST FROM AN XML RSS FILE

The main page of the website contains a list of the most recently-posted blogs. The blogs are contained in an RSS XML file and are extracted using a LINQ to an XML query, as shown in Listing 2-2. The query results are used later in the exercises in order to list the most current blogs on the ASP.NET MVC 4 homepage.

LISTING 2-2: LINQ to XML Query

```
public static IEnumerable<BlogList> GetLinks(XDocument doc)
{
IEnumerable<BlogList> list =
    (from item in doc.Elements("rss")
                     .Elements("channel")
```


www.it-ebooks.info

LISTING 2-2: *(Continued)*

```
                    .Elements("item")
            select new BlogList
            {
             Title = item.Element("title").Value,
              Url = item.Element("link").Value,
              Description = item.Element("description").Value
            }).Take(6);
        return list;
}
```

In the sample ASP.NET website, the XML file loads into the `BlogList` objects and binds to a `GridView` control embedded into the Default.aspx file. Although no one-to-one mapping exists between ASP.NET Gridview and ASP.NET MVC, you have several possibilities to cross this gap. Some are built in and some are third party. Options include:

- JqGrid, which is part of the jQuery library
- Telerik, which has some controls
- MvcContrib, a download from CodePlex
- The built-in WebGrid
- Razor

All examples in this chapter use Razor capabilities, therefore when you create the views, remember to select Razor (CSHTML) from the drop-down and not ASPX (C#). The following are the basic steps required to list the most recent blogs in the XML RSS file:

1. Add a public class called `BlogList` to the Models folder.
2. Add two methods to the Controllers\HomeControllers.cs file:
   - `BlogListXML()`
   - `GetLinks(XDocument doc)`
3. Modify the `Index()` method in the Controllers\HomeControllers.cs file to return the list of blogs.
4. Modify the Views\Home\Index.cshtml file to display the list of blogs to the view.

The following sections cover all these steps in greater detail.

Adding the BlogList Class to the Models Folder

To add the BlogList model, you right-click the Models folder and then select Add ➪ Class. Figure 2-3 illustrates the contents of the Add New Item window. You then name the class `BlogList.cs` and then click the Add button. Create the class as shown in Listing 2-3.

www.it-ebooks.info

The `BlogList` class and any, for that matter, class is a template that defines the form of an object. When the `BlogList` class is instantiated and data loaded into it, that data can be accessed for manipulation and presentation. The `BlogList` class contains the data presented in the list on the homepage.

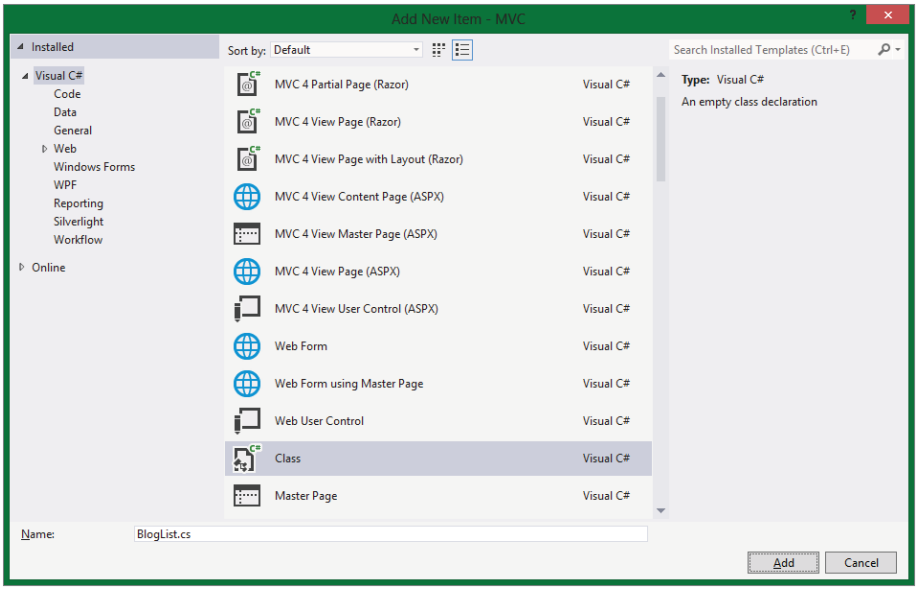


FIGURE 2-3

LISTING 2-3: The BlogList Class

```
public class BlogList
{
   public string Title { get; set; }
   public string Url { get; set; }
   public string Description { get; set; }
}
```

Adding Methods to the HomeControllers File

To add the `BlogListXML()` and `GetLinks(XDocument doc)` methods, you simply open the Controllers\HomeControllers.cs file and add the them as shown in Listings 2-4 and 2-5. If you take a closer look at Listings 2-5 and 2-2, you will notice they are the same code. In this case, you can reuse the code you used with the ASP.NET 2.0 application in the ASP.NET MVC 4 without any modification.

www.it-ebooks.info

The `BlogListXML()` method loads the XML file that contains the blogs, calls the `GetLinks()` method that executes the LINQ to XML query, populates the `List<BlogList>`, and returns the list.

LISTING 2-4: The BlogListXML() Method

```
using MVC.Models;
using System.Xml.Linq;

public List<BlogList> BlogListXML()
{
  XDocument doc = XDocument.Load(@"C:\...\MVC\MVC\Content\RSS\csharp2011.xml");
  IEnumerable<BlogList> list = GetLinks(doc);

  List<BlogList> resultList = new List<BlogList>();
  foreach (BlogList blog in list)
  {
    blog.Description = blog.Description.Substring(0, 120) + "...";
    resultList.Add(blog);
  }
  return resultList;
}
```

NOTE *The code in Listing 2-4 might fail if the file does not exist. In an example in Chapter 6 this actually occurs The previous code is changed to use a C# method for loading files using a relative path.*

The `GetLinks()` method (Listing 2-5) creates an `IEnumerable` list containing the `BlogList` class. The LINQ query restricts the result to six blogs as you can notice from the `.Take(6)` command. You want to do this to make the main page format look good. Leaving this command off results in the retrieval of all the blogs in the XML document, which fills up the default page.

LISTING 2-5: The GetLinks(XDocument doc) Method

```
public static IEnumerable<BlogList> GetLinks(XDocument doc)
{
  IEnumerable<BlogList> list = (from item in doc.Elements("rss")
                                                .Elements("channel")
                                                .Elements("item")
                                select new BlogList
                                {
                                  Title = item.Element("title").Value,
                                  Url = item.Element("link").Value,
```


www.it-ebooks.info

LISTING 2-5: *(Continued)*

```
                                Description = item.Element("description").Value
                            }).Take(6);
  return list;
}
```

Listing 2-6 is an example of the XML RSS file. Try and connect the `.Elements` selected in Listing 2-5 with the elements shown in the XML file, for example, `rss`, `channel`, `item`, `title`, `link`, and `description`.

LISTING 2-6: Example XML RSS File

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
  <channel>
    <item>
      <title>Using the as keyword versus boxing in C#</title>
      <description>
        When I convert one object to another using the keyword "as" and the
        original value is not of that type, the converted value simply becomes
        NULL.  For example, if theItem is of type MessageBox, then row will be
      </description>
      <link>
        http://www.thebestcsharpprogrammerintheworld.com/blogs/...boxing.aspx
     </link>
    </item>
  </channel>
</rss>
```

Modifying the Index() Method

Before you make modifications to the view (that is, View\Home\Index.cshtml), you must send the model from the controller. This is accomplished by adding a single line of code to the `Index()` method of the Controller\HomeController.cs and modifying the returned value. Listing 2-7 provides an example of how this method should look.

LISTING 2-7: Index() of the HomeController.cs

```
public ActionResult Index()
{
  ViewBag.Message = "This site is created to show … Windows Azure";
  var blogs = BlogListXML();
  return View(blogs);
}
```


www.it-ebooks.info

NOTE *A* `ViewBag` *is special to MVC and is a* `public dynamic` *property declared in the* `ControllerBase` *class, which implements the* `IController` *interface. When you declare something as* `dynamic`*, this means that the actual type is only known at run time and that you do not need to be concerned with the value stored in the* `ViewBag` *property. To view the definition, right-click* `ViewBag` *and then click Go To Definition.*

Displaying the List of Blogs in the View

Now that the contents of the XML file are loaded into a `List<BlogList>` object and is sent back to the view, it's time to bind that list to the view.

NOTE *If multiple models need to be returned to a view, use the* `ViewData[]` *collection that supports this requirement. This gets or sets a dictionary containing the data passed between a controller and the view.*

To do this, open the Views\Home\Index.cshtml file and add the following line of code as the first line:

```
@model List<MVC.Models.BlogList>
```

The loading and binding worked without that line of code because only a single model is returned. However the IntelliSense that you used later in the `foreach` loop was not enabled without it. So, you should add it because it makes things easier. Finally, add the HTML/Razor code to the Index.cshtml file, as shown in Listing 2-8, to display the list of blogs.

LISTING 2-8: Markup and Razor Code to Display the XML RSS Results

```
<table>
@foreach (var item in Model)
{
  <tr>
    <th>
      @item.Title
    </th>
  </tr>
  <tr>
    <td>
      @item.Description
    </td>
  </tr>
  <tr>
    <td>
```


www.it-ebooks.info

LISTING 2-8: *(Continued)*

```
      <a href="@item.Url">More...>>></a>
    </td>
  </tr>
}
</table>
```

When you press F5 to run the ASP.NET MVC application, you see the website rendered (refer to Figure 2-2).

ADDING THE HTML.ACTIONLINKS

You should have transformed the main page to how it needs to look (refer to Figure 2-2) and the page should contain the required functionality. If not, make the necessary changes based on your tastes, or copy the code from the downloadable ASP.NET MVC 4 example. The link to the source code is provided at the beginning of the chapter. The next step is to add links to the pages that take you to the default website page. For many of the example pages, a Model is not needed — only the controller and view.

Before the links are added to the default page (that is, the View\Home\Index.cshtml view) you must first create a page to link to. In this section, you create the `CSharpFundamentals` view and link to it from the View\Home\Index.cshtml view. To do so, follow these steps:

1. Add a controller to the MVC application by right-clicking the Controllers folder and then clicking Add ➪ Controller.
2. In the Add Controller window, add the contents shown in Figure 2-4, and click the Add button. In this figure, the Empty MVC Controller option is selected because the page you're linking to does not require any data retrieval or modification. If this example had used, for example ADO.NET Entity Framework, you could select the specific model and read\write can be dynamically generated. The `CSharpFundamentalsController` requires no modification because the page is basically static.
3. To add a view, right-click the `Index()` method of the CSharpFundamentalsController.cs file, and select Add Views, as shown in Figure 2-5.
4. In the Add view window, leave the default values as shown in Figure -2-6, and click the Add button. A folder called CSharpFundamentals is created in the Views folder automatically. It contains a file called Index.cshtml. This is the file where you place the static content to present to the visitor of the page.
5. Add the link from the main page (Home\Index.cshtml) to the Index.cshtml in the CSharpFundamentals view. You do this by adding the following code snippet to the Home\Index.cshtml file.

```
@Html.ActionLink("C# Fundamentals", "Index", "CSharpFundamentals")
```


www.it-ebooks.info

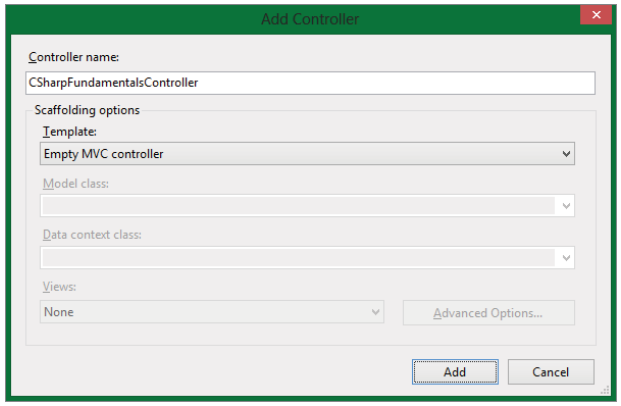


FIGURE 2-4

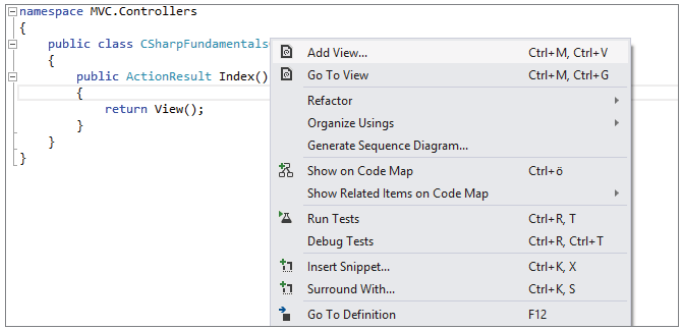


FIGURE 2-5

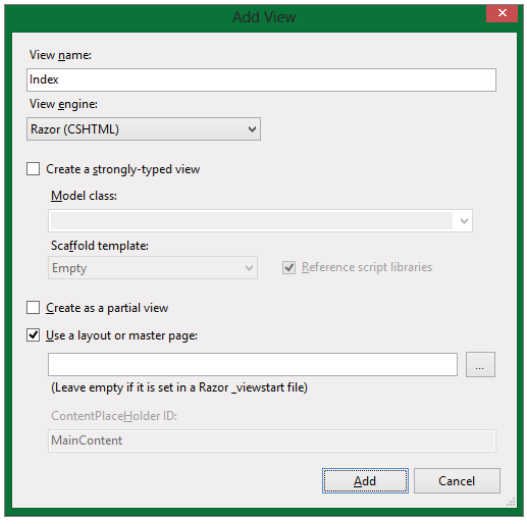


FIGURE 2-6

www.it-ebooks.info

NOTE *In the previous code snippet, the* `"C# Fundamentals"` *value is the text that appears on the Home\Index.cshtml, and it leads to the CSharpFundamentals\Index.cshtml page (that is, linkText). The* `"Index"` *is the* `actionName` *that matches the method found in the associated* `C#` *controller (that is,* `Index()`*).*

6. Press F5 to run the ASP.NET MVC 4 application.
7. Clicking the C# Fundamentals link renders the CSharpFundamentals/Index.cshtml view.

Recall that when the CSharpFundamentals view is created the defaults are left unchanged. Due to this, you use the Views_ViewStart.cshtml layout to render the CSharpFundamentals View. This feature is identical to the Master Pages concept in the ASP.NET platform. By having a default layout, you maintain the look and feel across the entire website. This makes it much easier to implement a look and feel change if required. For example, if you must change the size or color of the text in the `<table>` tag, making a change to the CSS class in the Site.css file impacts the entire website. Likewise, if you want to change the content shown from the _ViewStart.cshtml, you need to make the change only once. This is much better than having to access each web page and make a specific change to each.

Up to now the Home and C# Fundamentals pages have been migrated from ASP.NET 2.0 to ASP.NET MVC 4. This example website has two more pages containing only static text — the Blogs and Lessons page. You can use the process for migrating the C# Fundamentals and Home pages to migrate these other two static pages using the following actions:

- Add a Controller.
- Add a View.
- Add the `Html.ActionLink` to the Home\Index.cshtml.

The `Html.ActionLink()` methods implemented so far are presented in the following snippet. As you can see, there are links on the Home\Index.cshtml page for Home, Blogs, Lessons, and C# Fundementals, which all call the `Index()` method of their respective Controller. The name of the Controller is the last parameter of the `Html.ActionLink()` method parameter list.

```
@Html.ActionLink("Home", "Index", "Home")
@Html.ActionLink("Blogs", "Index", "Blogs")
@Html.ActionLink("Lessons", "Index", "Lessons")
@Html.ActionLink("C# Fundamentals", "Index", "CSharpFundamentals")
```

The remaining pages you're migrating to ASP.NET MVC 4 either contain a partial view that makes a database query or the page itself contains a database create, read, update, or delete (CRUD) action. Therefore, the next section describes how to implement NHibernate into the ASP.NET MVC 4 application.

www.it-ebooks.info

CREATING A LOCAL TEST DATABASE

The database part of this migration uses the Model-First design concept. This means that you must create the relationships between the classes as well as creating the classes themselves before creating the database tables and their foreign key relationships. NHibernate, like the Entity Framework, provides the capabilities for this. However, both NHibernate and Entity Framework require you first create a database in which to create the tables. In other words, the Object Relational Mappings (ORMs), such as NHibernate or Entity Framework, can create the tables and relationships between the tables for you, but they won't create the database itself.

NOTE *For more information on ORM's, see Chapter 1.*

Later chapters provide details for creating and migrating databases to the Windows Azure platform. In this example, however, because development happens locally on a PC, you create the database instance using Microsoft SQL Server 2012 Express. To create this local database, follow these basic steps, which are outlined in detail in the following sections:

1. Download and install Microsoft SQL Server 2012 Express (for example, SQLEXPR_x86_ENU.exe).
2. Create a new SQL Server Database using Visual Studio 2012.

Downloading and Installing SQL Server

To download and install SQL Server, you can do a search using any search engine for **SQL Server 2012 Express**. This likely renders the link to the Microsoft download site. There are many different versions of the download. You should choose the one appropriate to your platform and install the database on your computer.

NOTE *Although this example uses Microsoft SQL Server, you can use Oracle, MySQL, and so on by changing the drivers and connection strings.*

The installation process prompts the user with many options. To complete the installation process, follow these steps:

1. Choose the New SQL Server standalone option shown in Figure 2-7 to install the database. The Database Engine Configuration dialog appears (Figure 2-8).
2. Accept all the defaults except that shown in Figure 2-8. The plan is to use SQL Server authentication to control the access to the database. The other option is to allow Windows Authentication only, which is not what is used in this example. Be sure to write down the password for later use.

www.it-ebooks.info

New SQL Server stand-alone installation or add features to an existing installation
Launch a wizard to install SQL Server 2012 in a non-clustered environment or to add features to an existing SQL Server 2012 instance.

FIGURE 2-7

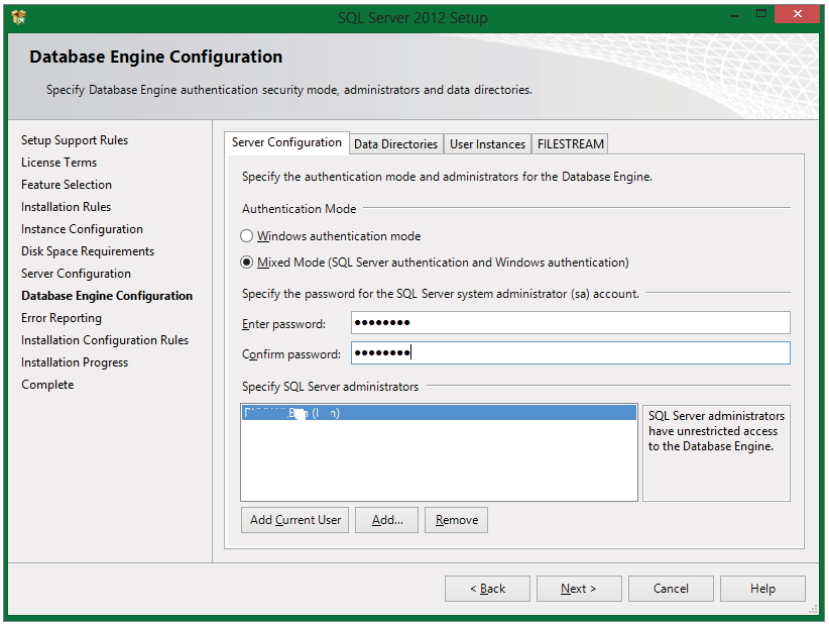


FIGURE 2-8

Creating a New SQL Server Database

To create the database, first, you need to open the Server Explorer in Visual Studio and then follow these steps:

1. Right-click Data Connections and then select Create New SQL Server Database, as shown in Figure 2-9. This opens the Create New SQL Server Database (Figure 2-10).

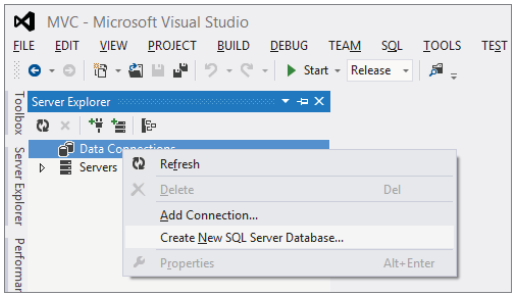


FIGURE 2-9

www.it-ebooks.info

2. Enter the values shown in Figure 2-10. The values include the Server Name, which it the name of your PC, because the database is being created locally followed by SQLExpress. For this example, you use SQL Server Authentication and create a username and password. The username and password can be anything you choose. Just remember them for later use. Lastly, enter the name of the database, in this case **Blog**.

NOTE *If you right-click the database you just created and select Properties (bottom right of the Visual Studio IDE), you can find the value of the Connection String property. This is useful as a guideline format to create the database Connection String in the application.*

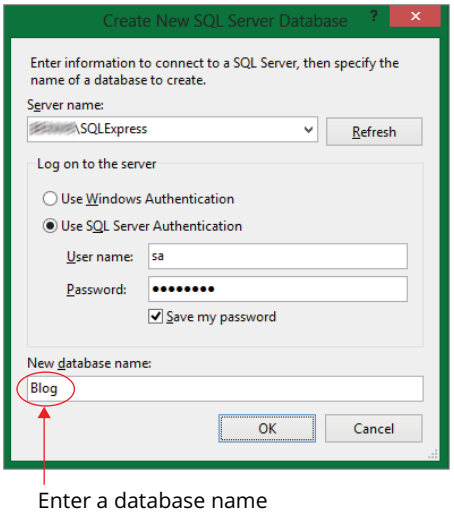


FIGURE 2-10

The database instance named Blogs is now created and ready for use. In the following section, you can find the instructions required to configure NHibernate, design the model, and create the tables are discussed in detail.

IMPLEMENTING NHIBERNATE INTO AN ASP.NET MVC 4 APPLICATION

This section illustrates how to perform an implementation using C# code. The actions required to implement NHibernate into an ASP.NET MVC 4 application follows:

www.it-ebooks.info

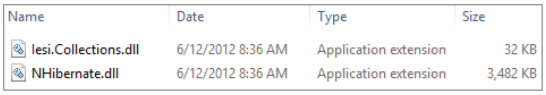
- Download, install, and add NHibernate references.
- Create the class and configuration mappings.
- Add the NHibernate configuration settings to the Global.asac.cs file. Download, and install NHibernate.

NOTE *My previous book,* Working with NHibernate 3.0, *has a chapter that covers implementing NHibernate into an ASP.NET MVC 3 application.*

You can find the most current version of NHibernate on the official website: `www.nhforge.org` or install it from the package manager in Visual Studio 2012. For this migration, version 3.3.1 is used.

The installation of NHibernate is relatively simple and can be completed in a number of ways:

- Copy the two dependent DLLs shown in Figure 2-11 into a directory on your PC. Then, in Visual Studio with the ASP.NET MVC 4 application open, right-click the References folder in the Solution window, and then click Add Reference. Browse to the location where you stored the two DLLs and add them to the solution.



Name	Date	Type	Size
Iesi.Collections.dll	6/12/2012 8:36 AM	Application extension	32 KB
NHibernate.dll	6/12/2012 8:36 AM	Application extension	3,482 KB

FIGURE 2-11

- A faster option is to use Visual Studio 2012's Library Package Manager. The Library Package Manager uses the NuGet package management system. With the ASP.NET MVC 4 application open, select Tools ➪ Library Package Manager ➪ Package Manager Console. This results in the Package Manager Console being rendered at the bottom of the Visual Studio IDE. Figure 2-12 illustrates this. You then enter **install-package NHibernate**, and the most current version of NHibernate installs into your project. In Figure 2-13, the same two DLLs as those shown in Figure 2-11 are installed into the References directory. In this case, the versions are actually newer than the ones downloaded from the NHibernate website.

NOTE *NuGet is an open source tool that is helpful for integrating third-party libraries directly into a .NET application.*

---

www.it-ebooks.info

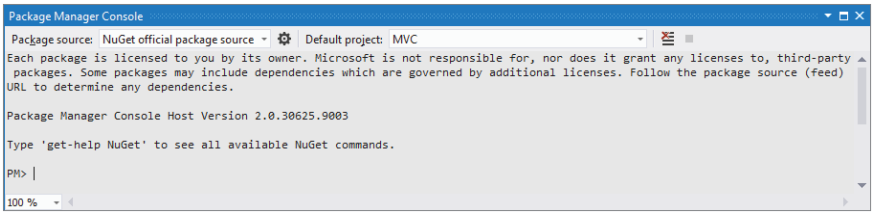

Package Manager Console
Package source: NuGet official package source
Default project: MVC
Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages. Some packages may include dependencies which are governed by additional licenses. Follow the package source (feed) URL to determine any dependencies.
Package Manager Console Host Version 2.0.30625.9003
Type 'get-help NuGet' to see all available NuGet commands.
PM>
100 %

FIGURE 2-12

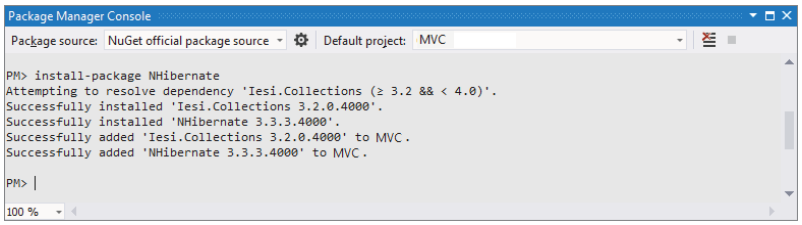

Package Manager Console
Package source: NuGet official package source
Default project: MVC
PM> install-package NHibernate
Attempting to resolve dependency 'Iesi.Collections (≥ 3.2 && < 4.0)'.
Successfully installed 'Iesi.Collections 3.2.0.4000'.
Successfully installed 'NHibernate 3.3.3.4000'.
Successfully added 'Iesi.Collections 3.2.0.4000' to MVC.
Successfully added 'NHibernate 3.3.3.4000' to MVC.
PM>
100 %

FIGURE 2-13

At this point, NHibernate is installed and ready for usage.

CREATING THE NHIBERNATE CLASSES AND CONFIGURATION

Now that NHibernate is installed, the remaining thing to do is add the `using{}` statement in the C# files to access its capabilities. To do this, you create a static class-level `SessionFactory`, a method that configures NHibernate called `ConfigureNHibernate()`, and the classes used in the ASP.NET MVC 4 application that need the database tables, Blog and Comments. Then implement the code that exports the schema, creates the database, validates the schema, and inserts the default data. To do all this, follow these steps:

1. Add a class-level `SessionFactory` to the Global.asax.cs file, as shown in Listing 2-9.

LISTING 2-9: Adding the SessionFactory to the Global.asax.cs File

```
using NHibernate;
namespace MVC
{
    public class MvcApplication : System.Web.HttpApplication
    {
        public static ISessionFactory SessionFactory { get; private set; }

        …
    }
}
```


www.it-ebooks.info

2. In the same file (Global.asax.cs) create a method called `ConfigureNHibernate()`, as shown in Listing 2-10. Remember that you can retrieve the value for the db.ConnectionString by right-clicking the Blogs database under the Database connections list.

LISTING 2-10: NHibernate configuration Method

```
using NHibernate.Cfg;
using NHibernate.Dialect;
using NHibernate.Driver;
using System.Data;
private static Configuration ConfigureNHibernate()
{
  var configure = new Configuration();
  configure.SessionFactoryName("MVC-TheBestCSharpProgrammerInTheWorld");

  configure.DataBaseIntegration(db =>
  {
    db.Dialect<MsSql2008Dialect>();
    db.Driver<SqlClientDriver>();
    db.KeywordsAutoImport = Hbm2DDLKeyWords.AutoQuote;
    db.IsolationLevel = IsolationLevel.ReadCommitted;
    db.ConnectionString = @"Data Source=BENW8\SQLExpress;
                            Initial Catalog=Blogs;Persist
                            Security Info=True;User ID=sa;
                            Password=**;Pooling=False";
    db.Timeout = 10;

    // enabled for testing
    db.LogFormattedSql = true;
    db.LogSqlInConsole = true;
    db.AutoCommentSql = true;
  });

  return configure;

}
```

3. Call the `ConfigureNHibernate()` method from within the `Application_Start()` method of the Global.asax.cs file, as shown in Listing 2-11. Add the method call to the end of the existing `Application_Start()` logic. The session context has been set to `WebSessionContext`.

www.it-ebooks.info

LISTING: 2-11: Executing the ConfigureNHibernate() Method

```
using NHibernate.Context;
var configuration = ConfigureNHibernate();
configuration.CurrentSessionContext<WebSessionContext>();
```

4. Create the classes and the relationship between the classes. These classes are what NHibernate ultimately uses to create the tables in the database. Add a class called `Blog` to the Models directory by right-clicking the folder; then click Add ➪ Class ➪ Blog.cs ➪ Add. The structure of the class is shown in Listing 2-12.

LISTING 2-12: The Blog Class

```
public class Blog
{
  public Blog()
  {
    comments = new List<Comments>();
  }

  public virtual int Id { get; set; }
  public virtual string Title { get; set; }
  public virtual string Type { get; set; }
  public virtual string FileName { get; set; }
  public virtual DateTime CreationDate { get; set; }
  public virtual string Category1 { get; set; }
  public virtual string Category2 { get; set; }
  public virtual string Category3 { get; set; }
  public virtual string Category4 { get; set; }
  public virtual Decimal Rating { get; set; }
  public virtual int NumberOfRatings { get; set; }
  public virtual int Advanced { get; set; }

  public virtual IList<Comments> comments { get; set; }
}
```

NOTE *At this point, you have a reference to the* `Comments` *class, which has not yet been created in this class. These* `Comments` *class references — for example in the* `Blog()` *constructor and as the type of elements in the* `IList` *— retrieve the comments for a specific blog. Just ignore the error until after the* `Comments` *class is created, as shown in Listing 2-13.*

5. Add another class to the Models directory by right-clicking the Models directory and then clicking AddClass ➪ Comments.cs ➪ Add. Listing 2-13 shows the structure of the `Comments` class.

www.it-ebooks.info

LISTING 2-13: The Comments Class

```
public class Comments
{
   public Comments() { }

   public virtual int Id { get; set; }
   public virtual string Subject { get; set; }
   public virtual string Comment { get; set; }
   public virtual string Email { get; set; }
   public virtual string Name { get; set; }
   public virtual string Password { get; set; }
   public virtual string Url { get; set; }
   public virtual DateTime TimeStamp { get; set; }
   public virtual string Type { get; set; }

   public virtual Blog blog { get; set; }
}
```

NOTE *Before the release of NHibernate 3.0, the mappings of the classes to a database table generally took place using mapping files. These files are XML-based and required that the developer manage the structure of the class and its relationship with the database in two different places — one in the class file and one in the mapping file. With the release of NHibernate 3.0, there came the concept of mapping by code that eliminated the need for the mapping/xml files. Nonetheless, using XML files to map your classes to a database is still supported and widely used.*

6. To map the `Blog` class, open the Blog.cs file and add the code in Listing 2-14. Include the `NHibernate.Mapping.ByCode` and `NHibernate.Mapping.ByCode.Conformist` namspaces — these provide the methods required to perform the mappings. Notice that the class contains a `Bag`. This custom container is the link between the `Blog` class and the `Comments` class. This relationship is what NHibernate uses to create the database relationship.

LISTING 2-14: The BlogMap Class

```
using NHibernate.Mapping.ByCode;
using NHibernate.Mapping.ByCode.Conformist;
public class BlogMap : ClassMapping<Blog>
{
  public BlogMap()
  {
    Id<int>(x => x.Id);
    Property<string>(x => x.Title);
    Property<string>(x => x.Type);
```


www.it-ebooks.info

LISTING 2-14: *(Continued)*

```
    Property<string>(x => x.FileName);
    Property<DateTime>(x => x.CreationDate);
    Property<string>(x => x.Category1);
    Property<string>(x => x.Category2);
    Property<string>(x => x.Category3);
    Property<string>(x => x.Category4);
    Property<Decimal>(x => x.Rating);
    Property<int>(x => x.NumberOfRatings);
    Property<int>(x => x.Advanced);

    Bag<Comments>(x => x.comments, cp => {},
                  cr => cr.OneToMany(x => x.Class(typeof(Comments))));
  }
}
```

7. As with the `BlogMap` class, add a `CommentsMap` class, as shown in Listing 2-15. Notice that instead of a custom `Bag` container, there is simply a `ManyToOne` relationship created of type `Blog`.

LISTING 2-15: The CommentsMap Class

```
public class CommentsMap : ClassMapping<Comments>
{
 public CommentsMap()
 {
   Id<int>(x => x.Id);
   Property<string>(x => x.Subject);
   Property<string>(x => x.Comment);
   Property<string>(x => x.Email);
   Property<string>(x => x.Name);
   Property<string>(x => x.Password);
   Property<string>(x => x.Url);
   Property<DateTime>(x => x.TimeStamp);
   Property<string>(x => x.Type);
   ManyToOne<Blog>(x => x.blog);
  }
}
```

8. Return to the Global.asax.cs file located in the root directory of the MVC solution, and add the method shown in Listing 2-16. The `GetMappings()` method loads the classes and compiles them so that they can be used to create the data structure.

www.it-ebooks.info

LISTING 2-16: The GetMappings() Method

```
using NHibernate.Cfg.MappingSchema;
using NHibernate.Mapping.ByCode;
protected static HbmMapping GetMappings()
{
 ModelMapper mapper = new ModelMapper();
 mapper.AddMapping<MVC.Models.BlogMap>();
 mapper.AddMapping<MVC.Models.CommentsMap>();
 return mapper.CompileMappingFor(new[] { typeof(MVC.Models.Blog),
                                         typeof(MVC.Models.Comments) });
}
```

9. Add the lines of code shown in Listing 2-17 to the `Application_Start()` method of the Global.asax.cs file. Add this code below the code added in Listing 2-11.

LISTING 2-17: Mapping by Code – NHibernate Methods

```
using NHibernate.Tool.hbm2ddl;
HbmMapping mapping = GetMappings();
configuration.AddDeserializedMapping(mapping, "ASP.NET.MVC.4");
SchemaMetadataUpdater.QuoteTableAndColumns(configuration);
SessionFactory = configuration.BuildSessionFactory();
```

At this point, everything is set up and ready to go from an NHibernate perspective. However, the first time or anytime you want to create the database schema, you must take some additional steps. These methods must run only once to initially create the database schema using the classes created previously. Listing 2-18 should be executed only when the database schema needs to be created, dropped, and/or re-created.

10. Add the code shown in Listing 2-18 to the Global.asax.cs file after the code added in Listing 2-17. This code snippet calls the `Drop()` and `Create()` methods of the `SchemaExport NHibernate` class.

LISTING 2-18: NHibernate Create, Validate, and Insert Data into Database Schema

```
//Comment this code out unless you want to build
//the database at the start of each application
try
{
  new SchemaExport(configuration).Drop(false, true);
  new SchemaExport(configuration).Create(false, true);
  SchemaValidator schemaValidator = new SchemaValidator(configuration);
  schemaValidator.Validate();
  //Insert some default data if required
```


www.it-ebooks.info

LISTING 2-18: *(Continued)*

```
}
catch (HibernateException e)
{
  //Log the error in the Event Viewer using the System Diagnostics library
}
```

WARNING *Be certain that you want to* `Drop()` *your database before you run this method because the method deletes everything. The* `SchemaValidator` *ensures everything works all right, and that your database schema and the mappings are the same.*

Figure 2-14 shows the status of the database before and after the running of the website.

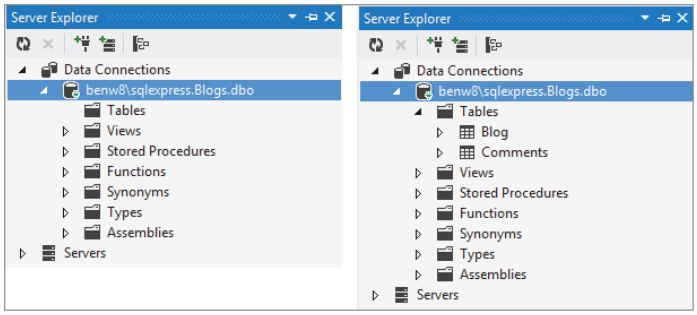


FIGURE 2-14

11. Expand the Comments table so that the columns are visible, and notice the column named Blog. This is actually a foreign key that makes the connection between the Blog and the Comments. NHibernate also marks the ID columns as the primary key for both the Blog and the Comments table.

In the downloadable sample website mentioned at the beginning of this chapter you can find some example data into the tables. To access them, you can search for the `InsertTestData()` method located in the Global.asax.cs file.

This completes the implementation of NHibernate into an ASP.NET MVC application. The next two sections give examples of using NHibernate in the application to retrieve data.

www.it-ebooks.info

CREATING AND ADDING THE BLOGNAVBAR PARTIAL VIEW

The BlogNavBar in this sample website is what contains the Translator, Blog Statistics (Counts and Archives), Disclaimer, Blog Archive, and so on. This navigation bar/partial view is part of many pages contained on the website. Every blog has this view added to the right side of the page. In addition, the .NET Fundamentals, Reviews, Archive List, Advanced C#, and so on pages have this partial view included. As discussed in Chapter 1, it makes sense to create a control or partial view and reuse it across the website instead of adding the code individually to each page.

The BlogNavBar contains two sections that require data retrieval:

- **Blog Statistics:** Contains the number of blogs, number of comments and the number of fundamental blogs.
- **Blog Archive List:** Contains a list of blogs by month and year along with the number of blogs for that month.

The following actions are required to complete the creation of the BlogNavBar:

1. Add the _BlogNavBar.cshtml partial view to the Views\Shared directory.
2. Add the static content to the view.
3. Create a sample view (netFundamentals.cshtml) and add the BlogNavBar partial view.
4. Add the logic to retrieve and present database content.

The sections further detail these general steps to get your partial view up and running.

Adding a Partial View

You can easily add the partial view to the _BlogNavBar.cshtml by following these steps:

1. Right-click the Views\Shared directory, and then select Add View. The Add View dialog appears.
2. Change the view name to **BlogNavBar**.
3. Select the Create as Partial View check box.
4. Click the Add button.

Figure 2-15 illustrates the values for the creation of this partial view.

www.it-ebooks.info

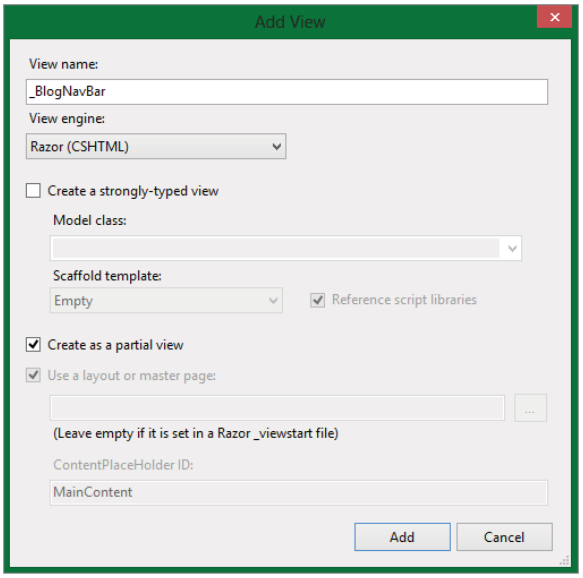


FIGURE 2-15

Adding Static Content to the View

Again, use the ASP.NET sample website code contains the static content for this partial view, or create your own content and structure of the BlogNavBar.

NOTE *The ASP.NET BlogNavBar control is found in the Include directory of the ASP.NET sample allocation, and the file is called BlogRightColumn.ascx. The sample ASP.NET website contained with the file named Chapter 2 - CSHARP.ASP.NET is downloadable from the Wrox website.*

Creating a Sample View and Adding the _BlogNavBar

Now that you've added static content to the BlogNavBar partial view, you can add it to a web page in the test ASP.NET MVC 4 website. You use the netFundamentals.aspx file from the sample ASP.NET website as a base for this next task and the content for the page. Follow these steps to convert and place this file into the ASP.NET MVC 4 application:

1. Add a new View to the Views\CSharpFundamentals directory called **netFundamentals.cshtml**. To do so, right-click the Views\ CSharpFundamentals directory, and select Add ➪ View. Then change the view name to **netFundamentals**, leaving all other values as default. Finally, click the Add button. The netFundamentals.cshtml file is then added to the directory.

www.it-ebooks.info

2. Modify the CSharpFundamentalsController.cs to add the required ActionResult. Add the method shown in Listing 2-19 to the Controllers\CSharpFundamentalsController.cs file.

LISTING 2-19: The Default ActionResult Method for the netFundamentals.cshtml View

```
public ActionResult netFundamentals()
{
  return View();
}
```

3. Add the ActionLink to the Views\Shared_Layout.cshtml to call the `ActionResult` method you added in step 2. Add the following code snippet to the Views\Shared_Layout.cshtml directly under the C# Fundamentals `Html.ActionLinks` added in the previous section "Adding the Html.ActionLinks."

```
@Html.ActionLink(".NET Fundamentals", "netFundamentals", "CSharpFundamentals")
```

4. Test that the link works and the netFundamental.cshtml file is opened. Press F5 in Visual Studio 2012, and, after the main page is rendered, click the .NET Fundamentals link on the top left of the web page, as shown in Figure 2-16. The default netFundamentals.cshtml page is rendered using the `_Layout.cshtml` view that keeps the look and feel of the website consistent.

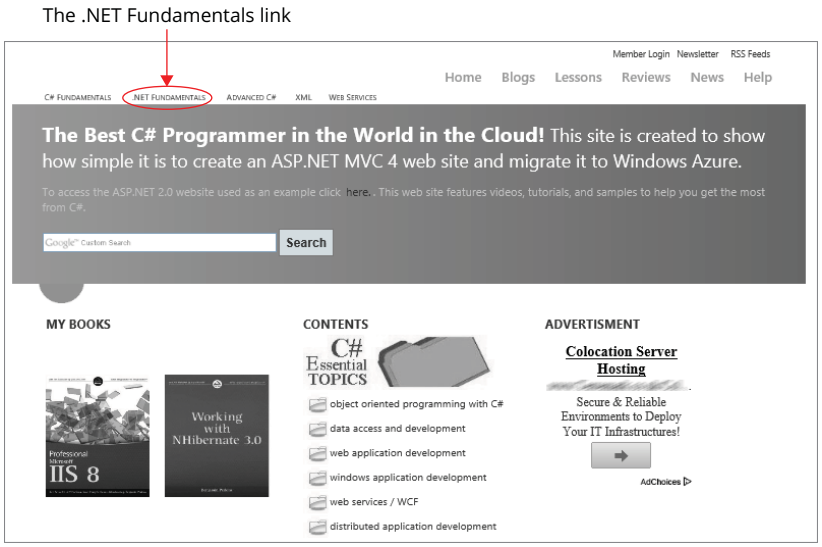

The .NET Fundamentals link
Member Login Newsletter RSS Feeds
Home Blogs Lessons Reviews News Help
C# FUNDAMENTALS .NET FUNDAMENTALS ADVANCED C# XML WEB SERVICES
The Best C# Programmer in the World in the Cloud! This site is created to show how simple it is to create an ASP.NET MVC 4 web site and migrate it to Windows Azure.
To access the ASP.NET 2.0 website used as an example click here.. This web site features videos, tutorials, and samples to help you get the most from C#.
Search
MY BOOKS
CONTENTS
C# Essential TOPICS
object oriented programming with C#
data access and development
web application development
windows application development
web services / WCF
distributed application development
ADVERTISMENT
Colocation Server Hosting
Secure & Reliable Environments to Deploy Your IT Infrastructures!
AdChoices

FIGURE 2-16

www.it-ebooks.info

5. Create the netFundamental.cshtml page. Use the netFundamentals.aspx page in the sample ASP.NET code.
6. Add the BlogNavBar partial view to the netFundamentals.cshtml. Migrate the content and look and feel to the new netFundamentals.cshtml page.
7. Add the dynamic content to the _BlogNavBar.cshtml partial view. Essentially, you add the View\Shared_BlogNavBar.cshtml partial view to the View\CSharpFundamentals\netFundamentals.cshtml page. This is done by adding the following code snippet to the View\CSharpFundamentals\netFundamentals.cshtml view:

```
@Html.Partial("_BlogNavBar")
```

There really is nothing significant to discuss with the creation of the static content for this file. It is simple HTML, tables, row, columns, some bulleted lists, and some content.

NOTE *The ASP.NET MVC 4 sample website can be downloaded from the Wrox website. If you need some tips or examples on how to create these pages, everything in these examples can be found there.*

---

8. Run the ASP.NET MVC 4 project. Select F5 and then to click the .NET Fundamentals links, as shown in Figure 2-16. The View\Shared_BlogNavBar.cshtml partial view renders along with the View\Shared_Layout.cshtml and the View\CSharpFundamentals\netFundamentals.cshtml view.

Adding Dynamic Content to the _BlogNavBar Partial View

In this section, you add the dynamic content that is retrieved from the database. In this example, the count of the total number of blogs, the count of the total number of comments, the count of the total number of fundamental articles, and the list of blog archives by `YEAR` and `MONTH` are the dynamic variables that need retrieval from the database. The following sections outline each database variable in greater detail.

Adding the Blog Archive by Year Month Partial View

The objective of this section is to extract the Year, the Month, and the number of blogs written during that time frame (that is, Year and Month) from the `blog` database table. The query shown in Listing 2-20 provides the required results.

www.it-ebooks.info

LISTING 2-20: Select Year, Month, and Count from the Blog Table

```
"SELECT DATENAME(YEAR, CREATIONDATE) AS YEAR, " +
"DATENAME(MONTH, CREATIONDATE) AS MONTH, COUNT(*) AS COUNT FROM Blog " +
"WHERE DATENAME(YEAR, CREATIONDATE) = DATENAME(YEAR, GetDate()) " +
"GROUP BY DATENAME(MONTH, CREATIONDATE), DATENAME(YEAR, CREATIONDATE) " +
"ORDER BY CASE LEFT(DATENAME(MONTH, CREATIONDATE), 3) " +
"WHEN 'JAN' THEN '01' " +
"WHEN 'FEB' THEN '02' " +
"WHEN 'MAR' THEN '03' " +
"WHEN 'APR' THEN '04' " +
"WHEN 'MAY' THEN '05' " +
"WHEN 'JUN' THEN '06' " +
"WHEN 'JUL' THEN '07' " +
"WHEN 'AUG' THEN '08' " +
"WHEN 'SEP' THEN '09' " +
"WHEN 'OCT' THEN '10' " +
"WHEN 'NOV' THEN '11' " +
"WHEN 'DEC' THEN '12' " +
"ELSE '' END DESC";
```

Two things to mention about this query:

- **It focuses only on the blogs written in the current year.** Each year on this blog, the blogs from the previous year were hard-coded. This was done for scalability reasons. Over time, if the projection of a 1-year time window was not implemented, the amount of retrieved data and the query execution time would increase. Therefore, it's a good idea to place this constraint on the query.
- **The result of this query is not strongly typed.** The query returns the year, month, and count, and there is not an existing class that has this structure, nor is there a NHibernate mapping class to connect it to. This means that the result of the query is not directly bound to a class and is therefore loaded into the type *object*.

You can begin the retrieval of this blog archive list by following these steps:

1. Create a class called `BlogNavBarElements`, as shown in Listing 2-12.
2. Add the class to the Models directory of the ASP.NET MVC 4 solution by right-clicking the Models directory and then clicking Add ➪ Class. Enter **`BlogNavBarElements`** in the Name text box and then click the Add button.

www.it-ebooks.info

LISTING 2-21: The BlogNavBarElements Class

```
public class BlogNavBarElements
{
  public int blogCount { get; set; }
  public int commentsCount { get; set; }
  public int fundamentalsCount { get; set; }
  public List<BlogArchives> blogArchiveList { get; set; }
}
```

This class is a hybrid class, meaning that it contains all dynamic content for the `BlogNavBar` partial view including the List<`BlogArchives`> collection.

3. The `BlogArchives` class is shown in Listing 2-22. Add it to the Models directory of the ASP.NET MVC 4 solution by right-clicking the Models directory and then clicking Add ➪ Class. Enter BlogArchives in the Name textbox and then click the Add button.

LISTING 2-22: The BlogArchives Class

```
public class BlogArchives
{
  public string Month { get; set; }
  public string Year { get; set; }
  public string Count { get; set; }
}
```

4. In the `Models\BlogNarBarElements` class add a method called `GetBlogNavBarContent()`, as shown in Listing 2-23.

   In this listing, the class type storing the return of `blogsQuery.List<object>` is of type `IList<object>` object. This is mentioned again to draw special attention that you cannot cast an untyped object directly to the Models\BlogArchives class (as shown in Listing 2-22). In Listing 2-23, the `IList<object> baList` is cast to a `List<string>` and then added to the `List<BlogArchives>` collection of the `Models\BlogNavBarElements` class.

NOTE *As a test, try to convert* `BlogNavBarElements` *directly from object to BlogArchives to see what happens. See if you can come up with a different or better way to implement this.*

---

www.it-ebooks.info

LISTING 2-23: Get the Year, Month, and Count of the Blogs for the Current Year

```
using NHibernate;

public BlogNavBarElements GetBlogNavBarContent()
{
  BlogNavBarElements elements = new BlogNavBarElements();
  string Hql = "SELECT DATENAME(YEAR, CREATIONDATE) AS YEAR, " +
  "DATENAME(MONTH, CREATIONDATE) AS MONTH, COUNT(*) AS COUNT FROM Blog " +
  "WHERE DATENAME(YEAR, CREATIONDATE) = DATENAME(YEAR, GetDate()) " +
  "GROUP BY DATENAME(MONTH, CREATIONDATE), DATENAME(YEAR, CREATIONDATE) " +
  "ORDER BY CASE LEFT(DATENAME(MONTH, CREATIONDATE), 3) " +
  "WHEN 'JAN' THEN '01' " +
  "WHEN 'FEB' THEN '02' " +
  "WHEN 'MAR' THEN '03' " +
  "WHEN 'APR' THEN '04' " +
  "WHEN 'MAY' THEN '05' " +
  "WHEN 'JUN' THEN '06' " +
  "WHEN 'JUL' THEN '07' " +
  "WHEN 'AUG' THEN '08' " +
  "WHEN 'SEP' THEN '09' " +
  "WHEN 'OCT' THEN '10' " +
  "WHEN 'NOV' THEN '11' " +
  "WHEN 'DEC' THEN '12' " +
  "ELSE '' END DESC";
  using (ISession session = MvcApplication.SessionFactory.OpenSession())
  using (ITransaction transaction = session.BeginTransaction())
  {
    IQuery blogsQuery = session.CreateSQLQuery(Hql);
    IList<object> baList = blogsQuery.List<object>();
    elements.blogArchiveList = new List<BlogArchives>();
    foreach (object[] archive in baList)
    {
      List<string> fields = archive.Select(i => i.ToString()).ToList();
      elements.blogArchiveList.Add(new BlogArchives
      {
        Year = fields[0].ToString(),
        Month = fields[1].ToString(),
        Count = fields[2].ToString()
      });
    }
  }
  return elements;
}
```

5. Add the code to the `netFundamentals()` method of the `Controllers\CSharpFundamentalsController` controller, as shown in Listing 2-24. This method creates an instance of the `Models\BlogNavBarElements` class and calls the `GetBlogNavBarContent()` method. The result of the method is stored in a `ViewData` object named `blognavbar`, which is returned to the view for presentation.

www.it-ebooks.info

LISTING 2-24: Return the Year, Month, and Count of the Blogs for the Current Year Using ViewData

```
BlogNavBarElements blogElements = new BlogNavBarElements();
ViewData["blognavbar"] = blogElements.GetBlogNavBarContent();
return View(ViewData);
```

Before making the necessary changes to the Views\Shared_BlogNavBar.cshtml partial view, finish the `GetBlogNavBarContent()`. The next section discusses the final modifications to this method.

Adding Total blogs, Total Comments and Total Fundamentals

Retrieving and formatting the Blog Archives were a little challenging. The conversion from object to string to BlogArchive took some effort. If you have made it through this chapter, the remaining three code listings should not be a problem.

1. The first code segment, which you add to the `GetBlogNavBarContent()` method of the `Models\BlogNavBarElements` class, is shown in Listing 2-25. It retrieves and stores the total number of blogs.

LISTING 2-25: Total Blog Count Query

```
using (ISession session = MvcApplication.SessionFactory.OpenSession())
using (ITransaction transaction = session.BeginTransaction())
{
  IQuery blogQuery = session.CreateQuery("Select count(*) from Blog");
  elements.blogCount = Convert.ToInt32(blogQuery.UniqueResult());
}
```

2. Add the following code segment shown in Listing 2-26. It retrieves and stores the total number of comments.

LISTING 2-26: Total Comments Count Query

```
using (ISession session = MvcApplication.SessionFactory.OpenSession())
using (ITransaction transaction = session.BeginTransaction())
{
  IQuery commentsQuery = session.CreateQuery("Select count(*) from Comments");
  elements.commentsCount = Convert.ToInt32(commentsQuery.UniqueResult());
}
```

3. Add the code shown in Listing 2-27 to the `GetBlogNavBarContent()` method of the `Model\BlogNavBarElements` class to retrieve and store the total count of articles of type `'fundamentals'`.

www.it-ebooks.info

LISTING 2-27: Total Fundamentals Count Query

```
using (ISession session = MvcApplication.SessionFactory.OpenSession())
using (ITransaction transaction = session.BeginTransaction())
{
  IQuery fundamentalsQuery = session.CreateQuery(
                               "Select count(*)
                                from Blog
                                where Type = 'fundamentals'");
  elements.fundamentalsCount = Convert.ToInt32(fundamentalsQuery.UniqueResult());
}
```

That's it. Now that all the dynamic content has been retrieved from the database and has been passed to the view, it's time to access it and display it within the view.

Displaying the Dynamic Data

The `BlogNavBar` partial view contains a number of Dynamic data that require database access and data retrieval. This section explains how to access the methods just written so that the information and data provided by them is presented to the website visitor.

1. Add the following code snippet to line 1 of the Views\Shared_BlogNavBar.cshtml partial view:

```
@model MVC.Models.BlogNavBarElements
```

2. Add the following code snippets to the location where the total number of blogs, total number of comments and total number of fundamentals are displayed in the `BlogNavBar`. You add the code snippets to the View\Shared_BlogNavBar.cshtml file.

```
Blogs:        @Html.ValueFor(b => b.blogCount)
Comments:     @Html.ActionLink(Model.commentsCount.ToString(), "Comments", "Blogs")
Fundamentals: @Html.ActionLink
              (Model.fundamentalsCount.ToString(), "Index", "CSharpFundamentals")
```

NOTE *The* `Comments` *action* `ID` *has not been created at this point, and you will receive an error message if the project is run and the link is selected. You can find the view in the Views\Blogs directory and the code handling the request in the Controllers\BlogsController .cs file within the* `Comments()` *method.*

---

3. Add the blog archive list using the code shown in Listing 2-28 to the _BlogNavBar .cshtml file.

www.it-ebooks.info

LISTING 2-28: Display Dynamic Blog Archive to the BlogNavBar Partial View

```
@foreach (MVC.Models.BlogArchives blogArchive in Model.blogArchiveList)
{
 <table>
  <tr>
   <td>
     @blogArchive.Year
     @blogArchive.Month
     (@Html.ActionLink(blogArchive.Count, "ArchiveList", "Blogs"))
   </td>
  </tr>
 </table>
}
```

> **NOTE** *The* `ArchiveList` *view is created in the next section. If you click the link created in Listing 2-27, you will receive an error until the next few steps are completed.*

The previous listing segment loops through the `List<BlogArchives> blogArchiveList` collection located in the `BlogNavBarElements` class. The sample data adds to the sample database when the code in the Global.asax.cs file is uncommented. It sets the creation date to two months in advance and two months previous to the current date. Therefore, the provided sample data always includes four months of data. Figure 2-17 illustrates an example of the populated `blogArchiveList` collection in Visual Studio 2012 running in debug mode.

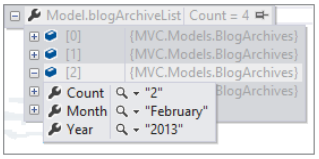


FIGURE 2-17

The `BlogNavBar` partial view is complete. You can now include it into any of the current web pages or to future additions using the following code snippet. The code has been extended to include the `ViewData` parameter, which was not present the first time you used this code snippet for testing (refer to Listing 2-24).

```
@Html.Partial("_BlogNavBar", ViewData["blognavbar"])
```

In the following sections, you can add the `BlogNavBar` to the Archive List web page, and also create an example blog post.

www.it-ebooks.info

NOTE *The creation of the Reviews, News, Help, Comments, and Advanced C# web pages are not discussed here. They are, however, part of the downloadable ASP.NET MVC 4 sample website (Chapter 2 - MVC.zip) available on Wrox.*

CREATING THE ARCHIVE LIST WEB PAGE

The Archive List page is the page that renders the list of blogs written during a selected year and month. Recall from the _BlogNavBar.cshtml partial view that the count value of the dynamic data in the `blog.Archive()` section is a link to the Archive List. The Archive List receives the year and month as parameters, which are then used as constraints in the NHibernate database query.

To migrate the Archive List ASP.NET web page to ASP.NET MVC 4, you need to perform the following actions (discussed in greater detail in the following sections):

- Create the ArchiveList.cshtml view in the View\Blogs directory.
- Add the `ArchiveList()` method to the Controllers\BlogController.cs file.
- Create and implement a custom `MapRoute`.
- Use LINQ to NHibernate to retrieve strongly typed data from the database using the provided parameters.
- Use `ViewData` and `ViewBag` containers to store and return the data to the view.
- Modify the ArchiveList.cshtml and_BlogNavBar.cshtml views to display and link to the data.

Creating the ArchiveList.cshtml View

The `ArchiveList` if the page that displays the list of blogs matching the Year and Month parameters when being accessed from the BLogNavBar or which match the Cloud Tag value. For example, if NHibernate is selected from the tag cloud, then all blogs written about NHibernate is retrieved from the database and presented to the website visitor.

To add a new view named ArchivetList.cshtml, follow these simple steps:

1. Right-click the Views\Blogs directory, and then click Add ➪ View.
2. In the Add View window, name the view `ArchiveList`, leaving the remaining default setting as they are.
3. Click the Add button.

www.it-ebooks.info

If you have downloaded the example code (found in the Chapter 2 - MVC.zip file), you can look at some of the other views and copy/paste much of the format for this view. For example, the `@section featured` is standard across the entire website and can be copied from any of the existing views.

Adding the ArchiveList() Action Result Method

From a manageability and maintenance perspective, it is a good idea to group related source files into better-structured subdirectories. Although the following is a small example, if you write a larger website, adding all your models, views, and controllers to a single directory would soon become unmanageable. Therefore, this example attempts to place everything directly related to a blog into the View\Blogs directory and the Controllers\BlogsController.cs file. To do so, follow these steps:

1. Add the `ArchiveList()`to the Controllers\BlogControllers.cs file, as shown in Listing 2-29.

LISTING 2-29: The ArchiveList() Action Result Method

```
public ActionResult ArchiveList()
{
  return View();
}
```

Now is a good time to test and make sure that the ArchiveList.cshtml view is how you want to style it and that the `ArchiveList()` method is wired correctly.

2. Select F5 and navigate to a page where the BlogNavBar is visible.
3. Scroll down to the `blog.Archive()` section, and click one of the counts in the archive list. Figure 2-18 illustrates what you are looking for.

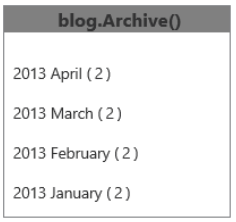


FIGURE 2-18

Create and Implement a Custom MapRoute

To implement the MapRoute, follow these steps:

1. Open the Global.asax.cs file, and add a method called `RegisterRoutes()`, which resembles that shown in Listing 2-30.

www.it-ebooks.info

LISTING 2-30: The Custom MapRoute for the ArchiveList Link

```
public static void RegisterRoutes(RouteCollection routes)
{
  routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
  routes.MapRoute(
      "Blogs",
      "Blogs/ArchiveList/{Year}/{Month}",
      new { controller = "Blogs", action = "ArchiveList", Year = "", Month = "" }
      );
}
```

The MapRoute creates a map that looks for a pattern like *Blogs/ArchiveList/2014/July*, for example.

2. To activate the custom MapRoute, add the following code snippet to the `Application_Start()` method also located in the Global.asax.cs file. A good place to put it is directly after the default generated code and before the previously added code that configured NHibernate.

```
RegisterRoutes(RouteTable.Routes);
```

Retrieving the Archive Blog Data with LINQ to NHibernate

The query in this section extracts the blogs written in the selected time frame. Because the result of this query is mapped directly to the `Blog` class, meaning it is strongly typed, a LINQ query is an option, unlike when the query returns an untyped result.

There are two actions required to extract the data and return it to the view for presentation. Both are performed in the BlogsController.cs file:

- Create a method called `GetArchiveList()` that executes the LINQ to NHibernate query.
- Modify the `ArchiveList()` action result method to return the data to the view.

To retrieve the archive blog data, follow these steps:

1. Open the Controller\BlogsController.cs, and create a method called `GetArchiveList()`, as shown in Listing 2-31. The method creates the NHibernate session and transactions, and then uses LINQ to NHibernate to extract the blogs from the database that matches the constraints in the `where` clause.

www.it-ebooks.info

LISTING 2-31: The GetArchiveList() Method

```
using NHibernate.Linq;

public IList<Blog> GetArchiveList(int Year, int Month)
{
  using (ISession session = MvcApplication.SessionFactory.OpenSession())
  using (ITransaction transaction = session.BeginTransaction())
  {
    IList<Blog> blogs = (from b in session.Query<Blog>()
                         where b.CreationDate.Year == Year &&
                               b.CreationDate.Month == Month
                         select b).ToList<Blog>();

    return blogs;
  }
}
```

NOTE *Don't forget to add the following statement:* `using NHibernate.Linq;`.

2. In the same Controller\BlogsController.cs file, modify the existing `ArchiveList()` method so that it resembles that shown in Listing 2-32.

LISTING 2-32: The Updated ArchiveList() Method That Includes Year and Month Parameters

```
public ActionResult ArchiveList(string Year, string Month)
{
  int iYear = Convert.ToInt32(Year);
  int iMonth = GetMonth(Month);

  ViewBag.Year = Year;
  ViewBag.Month = Month;

  ViewData["blogarchives"] = GetArchiveList(iYear, iMonth);
  BlogNavBarElements blogElements = new BlogNavBarElements();
  ViewData["blognavbar"] = blogElements.GetBlogNavBarContent();
  return View(ViewData);
}
```

The first action taken in this method is to convert the `Year` and `Month` into integers. This is required because the `b.CreationDate.Year` and `b.CreationDate.Month` in Listing 2-30 are integers and, therefore, need to match types.

www.it-ebooks.info

NOTE *The* `GetMonth()` *method is not shown here. However, it is a* `SWITCH` *statement that checks the month name and changes it to the number of that month.*

Next, the `ViewBag` container is used to pass the `Year` and `Month` passed into this method. They are passed back to the view and are used in the presentation of the archive list. Lastly, a call to the `GetArchiveList()` and the `GetBlogNavBarContent()` methods to populate the ViewData containers is performed.

Modifying the Views to Display and Link to Data

To finish up this section, the last action is to modify the `ArchiveList` and `BlogNavBar` views and present the data. You can do this by following these steps:

1. Modify the `Html.ActionLink` in the Views\Shared_BlogNavBar.cshtml file, added previously in Listing 2-28 so that is resembles the code shown in Listing 2-33.

LISTING 2-33: The _BlogNavBar partial view custom MapRoute link

```
@blogArchive.Year @blogArchive.Month (@Html.ActionLink(blogArchive.Count,
                                                        "ArchiveList",
new { controller = "Blogs",
      action = "ArchiveList",
      Year = @blogArchive.Year,
      Month = @blogArchive.Month
}))
```

2. Add the code shown in Listing 2-34 to the Views\Blogs\ArchiveList.cshtml below the `@` section feature added previously.

LISTING 2-34: Present the Data in the ArchiveList.cshtml View

```
<table>
 <tr>
  <td style="vertical-align: top; width: 900px">
   <table>
    <tr>
     <th>Archive List for @ViewBag.Month - @ViewBag.Year</th>
      </tr>
      <tr><td> </td></tr>
      @foreach (var item in ViewData["blogarchives"]
                                    as IEnumerable<MVC.Models.Blog>)
      {
      <tr>
       <td>
        <a href="http://www.thebes..eworld.com/@item.Type/@item.FileName">
```


www.it-ebooks.info

LISTING 2-34: *(Continued)*

```
                                                    @item.Title</a>
        </td>
       </tr>
       }
      </table>
     </td>
     <td style="vertical-align: top">
           @Html.Partial("_BlogNavBar", ViewData["blognavbar"])</td>
     </tr>
</table>
```

Now you can run the ASP.NET MVC 4 project, and click the links to see the list of blogs for that time period rendered as hyperlinks. The Archive List is complete. Take a breath and a break. Then continue on with the next section where a sample blog is migrated from ASP.NET to ASP.NET MVC 4.

MIGRATE A BLOG ENTRY FROM ASP.NET WITH FEEDBACK FORM AND COMMENT LIST

It is true that storing the blog contents in the database and retrieving them dynamically would be a little easier than having to create a blog page per blog. This was for Search Engine Optimization (SEO) reasons. When the search engine robots browse your website, they do not get access to all your content in your database. Therefore, creating a single search engine optimized page per blog gets more visibility to your blog. Plus, it makes you implement things such as Master Pages, partial views, and user controls, which are fun and useful.

The steps to migrate a blog from the ASP.NET sample website to an ASP.NET MVC 4 page follow:

- Create the blog view and add the static content.
- Add a generic action result method to the `Controller\BlogController.cs` file called `BlogLocator`.
- Create a method to retrieve the details of the blog from the database using the ID.
- Update the sample blog link to `\Blogs\BlogLocator\{id}`.
- Create a shared partial view for the blog comments and existing comment list.
- Add a partial view to the sample blog.

The example blog for this ASP.NET MVC 4 sample website is titled "Using the as keyword versus boxing." To create the view for this blog, do the following:

www.it-ebooks.info

1. Right-click the Views\Blogs directory, and then click Add ➪ View.
2. For the view name enter **Using-the-as-keyword-versus-boxing** and then click the Add button.
3. As you did with other static content for the web pages in this example, open one of the other files and copy/paste the content.
4. After you add the static content, add the reference to the View\Shared_BlogNavBar .cshtml partial view and continue to the next section, as shown in the following code snippet.

```
@Html.Partial("_BlogComments")
```

Adding a Controller to Manage Requests to All Blogs

Although there is only a single blog provided for this example, the logic created for the following exercise supports *n* number of blogs. This is because you use the Controller created in the Controller\BlogController.cs file for all blogs hosted on the website. To manage requests, add the method shown in Listing 2-35 to the Controller\BlogController.cs file.

LISTING 2-35: A Generic Controller action Method

```
public ActionResult BlogLocator()
{
  BlogNavBarElements blogElements = new BlogNavBarElements();
  ViewData["blognavbar"] = blogElements.GetBlogNavBarContent();
  return View(ViewData);
}
```

Creating a Method to Retrieve Blog Details

The code shown in Listing 2-34 currently contains only the logic to populate and return the dynamic content required for the BlogNavBar partial view. Now make changes to the method from Listing 2-34, and add an additional method to dynamically retrieve the selected blog. Listing 2-36 shows the method you use to retrieve the blog from the database using the ID.

LISTING 2-36: The GetBlogDetails Method

```
public Blog GetBlogDetails(int Id)
{
  using (ISession session = MvcApplication.SessionFactory.OpenSession())
  using (ITransaction transaction = session.BeginTransaction())
  {
    return session.Get<Blog>(Id);
  }
}
```


www.it-ebooks.info

To retrieve blog details, add the `GetBlogDetails()` method to the Controller\BlogController.cs file. When added, modify the `BlogLocator()` method shown previously in Listing 2-34 so that it resembles that shown in Listing 2-37.

LISTING 2-37: The Updated BlogLocator ActionResult Method

```
public ActionResult BlogLocator(int Id)
{
  Blog blog = GetBlogDetails(Id);

  BlogNavBarElements blogElements = new BlogNavBarElements();
  ViewData["blognavbar"] = blogElements.GetBlogNavBarContent();
  return View(blog.FileName, ViewData);
}
```

The changes to the `BlogLocator()` method shown in Listing 2-37 include

- Adding a parameter to the method called `Id`. The `Id` will be part of the URL, for example *~/Blogs/BlogLocator/100*, where *100* is the ID of the blog.
- Adding a call to the `GetBlogDetails()` method, which passes the `Id`.
- Modifying the parameters of the `View()` method to include the View name. In this example, the name is *Using-the-as-keyword-versus-boxing*, which matches the name of the View\Blogs Using-the-as-keyword-versus-boxing.cshtml view and is also the value of the *`blog.FileName`* property.

Updating the Example Blog Link

Recall from previous section "Modify the ArchiveList and BlogNavBar Views to Display and Link to the Data" that the links to the blogs were dynamically generated based on the details of the `List<Blog>` collection. Now return to the Views\Blogs\ArchiveList.cshtml view, and modify the `<a href>` value within the `foreach` statement so that is resembles the following code snippet.

```
<a href="/Blogs/BlogLocator/@item.Id">@item.Title</a>
```

`Blogs` is the controller, `BlogLocator` is the action, and the `@item.Id` is the ID. When the page is generated and the link is clicked, the request is routed to the correct view, based on the Id and the data retrieved from the database.

This is a good point to run the project and test if the previous methods and views work. After the tests are successful, move to the next section where adding a feedback form and a list of blog comments is covered.

www.it-ebooks.info

Creating a Shared Partial View for the Blog

For this example, the form that enables visitors to add a comment to the blog and the comments themselves are published on each blog. Instead of performing a cut and paste of the code onto each blog, this is another good situation for using a partial view. The partial view enables the creation of the view once and then it can be included into any other required view. To create a shared partial view, follow these steps:

1. To create a partial shared view, right-click the Views\Shared directory, and then click Add ➪ Views. The Add View dialog appears.
2. Add _BlogComments as the view name, and select the Create as a Partial View check box and click the Add button.

   In this example, the contents of the partial view are the form that enables a visitor to add a comment and the list of previously added comments. Listing 2-38 illustrates the most important code segments contained in the Views\Shared_BlogComments.cshtml file.

NOTE *The code to insert the comment into the database is not provided in this example or in the sample ASP.NET MVC 4 website. After you complete this chapter, try to modify the code and perform the action yourself.*

LISTING 2-38: Code Segment to List the Comments for a Specific Blog

```
@model IList<MVC.Models.Comments>
<table>
 @foreach (var comment in ViewData["comments"] as IEnumerable<MVC.Models.Comments>)
{
 <tr>
  <td><b>@comment.Subject</b> - @comment.Comment</td>
 </tr>
 }
</table>
```

3. Add the code shown in Listing 2-39 to the Controller\BlogController.cs file. The code retrieves all the comments for a given blog ID.

LISTING 2-39: The GetCommentsList Method with Id

```
public IList<Comments> GetCommentsList(int Id)
{
  using (ISession session = MvcApplication.SessionFactory.OpenSession())
  using (ITransaction transaction = session.BeginTransaction())
  {
```


www.it-ebooks.info

LISTING 2-39: *(Continued)*

```
    Blog blog = session.Get<Blog>(Id);
    IList<Comments> comments =  blog.comments.ToList<Comments>();
    return comments;
  }
}
```

> **NOTE** *Developers who know about NHibernate's Lazy Loading capabilities is probably asking why it was not used in this example. Unfortunately, due to the stateless nature of ASP.NET MVC and ASP.NET for that matter, the use of that capability in this context is not possible. The NHibernate implementation for this example is "Session per Database Transaction" and therefore there is no persisted session.*

4. Add the following code snippet to the `BlogLocator()` method in the Controller\BlogController.cs, as shown previously in Listing 2-34, before the View is returned.

```
ViewData["comments"] = GetCommentsList(Id);
```

Adding Partial View to the Blog

The final action for this chapter is to add the View\Shared_BlogComments.cshtml partial view to the sample blog. To do so, follow these steps:

1. Open the sample blog located in the Views\Blogs directory called Using-the-as-keyword-versus-boxing.cshtml.
2. Navigate to a location under the content, and add the following code snippet.

```
@Html.Partial("_BlogComments", ViewData["comments"])
```

3. Run and test the ASP.NET MVC 4 project, clicking the first blog on the list and viewing the form and comments. You can also navigate to any page linking to the Archive List. On the Archive List the link to this sample blog is hard-coded.

SUMMARY

This chapter covered a lot of topics, technologies, and features. You started with an ASP.NET website using ADO.NET and ended with an ASP.NET MVC 4 project using NHibernate. That is a significant change and something you can be proud of after you complete it.

www.it-ebooks.info

The most important aspects to remember from this chapter are that when you change, change as much as reasonably possible so that you not only make advancements in technology, but you also make advancements in your product and innovate. When this chapter was started, a decision was made to keep the same look and feel in the ASP.NET MVC 4 project as existed on the ASP.NET website. After some effort, the decision was made to change the look and feel of the website to be more like the ASP.NET MVC 4 project. By doing this, it not only updated the website from a technical perspective but also from a product perspective.

From a technical perspective, the configuration and implementation of NHibernate and the concepts around ViewData, ViewBag and how you use them to pass the models from the controller to the view are an important learning point for this chapter. Also, the creation of custom MapRoutes and partial views, which are similar to the ASP.NET user controls, give you a better grasp of what you can do with an ASP.NET MVC 4 website.

In the next chapter, you receive a review of the ASP.NET website and ASP.NET MVC 4 project from a performance perspective and gain some tips on how to optimize them for performance.

www.it-ebooks.info

PART II
Enhancing

- **CHAPTER 3:** Understanding ASP.NET MVC 4 Performance Optimization Techniques
- **CHAPTER 4:** Fine-tuning the ASP.NET MVC 4 Project for Performance

www.it-ebooks.info

3 Understanding ASP.NET MVC 4 Performance Optimization Techniques

CONCEPTS

IN THIS CHAPTER

- How to set a performance baseline
- Online tools for performance testing and optimization tips
- All about bundling and minification
- How to scale a Windows Azure Cloud Service
- How to improve performance

You can find some very interesting statistics about customer behavior based on a website's performance and responsiveness on the Internet. This includes the fact that more than 40 percent of website visitors will leave your website if the page takes more than three seconds to render. And that, in most cases, visitors to a website expect the page to render in two seconds or less. Also, 65 percent of shoppers dissatisfied with the performance of a website will never return, and will go someplace else next time. Performance of a website is one of the most important reasons visitors remain loyal (see Figure 3-1).

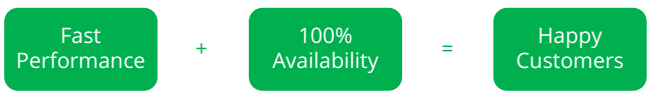


FIGURE 3-1

The normal human being can notice an event that takes approximately 500 ms or one-half a second, and will not notice anything that happens in less time than that. However, according to Amazon.com, an increase in the response of Amazon's system by as little as 100 ms results in a 1 percent loss in sales. In dollar terms, that's in the millions. Yahoo! also found that an increase of load time by 400 ms resulted in a 5–9 percent drop in its traffic.

www.it-ebooks.info

Customers and visitors will not use your site if the performance is poor. It's that simple. This chapter discusses some ways you can optimize your system for performance, and keep your customers or users happy and returning. Some points discussed in this chapter:

- The importance of setting a baseline
- Fiddler, F12 developer tools, and MiniProfiler
- Online tools for measuring performance and compatibility
- Bundling and minification
- Scaling a Windows Azure Cloud Service
- Fifteen ASP.NET and ORM performance enhancing tips

NOTE *This chapter contains key ASP.NET MVC optimization concepts. For a step-by-step guide to implement these concepts, see Chapter 4.*

SETTING A PERFORMANCE BASELINE

How do you know if your site has a performance problem? When someone mentions your website performs slowly, how do you determine if it really is? If you do not capture performance metrics, you'll spend a lot of time in meetings and on the phone explaining that the experienced performance is within the normal range. Without documented trends and metrics, both you and your customer will debate performance based on the perceived response times.

NOTE *You must capture performance statistics. You need to know when there is a performance problem and fix it before the user notices it.*

A number of tools are available to help you manually measure performance of your system. You can use one or all these tools to capture, store, and report the performance metrics of your system:

- **Fiddler:** Commonly used for troubleshooting issues between a client and a server, but also includes very valuable performance statistics.
- **Internet Explorer's F12 Developer Tools suite:** The suite provides a very nice graphical representation of all HTTP requests. You can export the data to Excel for further graphing and analysis.

www.it-ebooks.info

- **MiniProfiler:** A useful tool for tracking performance in your system down to a low level. It is simple to configure, install, and embed into your application. MiniProfiler supports ASP.NET, ORMs, ASP.NET MVC, and database query analysis.

Although you have tools and solutions to gather performance metrics automatically, they do have an impact on performance. Before you implement an automated solution for this, make sure you know how much impact it has on the system.

USING ONLINE TOOLS FOR PERFORMANCE TESTING AND OPTIMIZATION TIPS

A recently published website that provides some useful tips for ASP.NET optimization can be found at `http://webdevchecklist.com/asp.net/`. The website provides links to tools categorized into topics such as Security, Code Quality, Mobile, Performance, and so on. The checklist has a good set of tasks to ensure your website is ready to go into a live environment.

The Performance category for this site contains a number of actions for optimizing an ASP.NET website for performance, for example:

- Run in release mode (`debug="true"`)
- Google PageSpeed score of 90+
- Yahoo! YSlow score of 85+
- Optimize HTTP headers
- Optimize images

In Chapter 4 you use Google PageSpeed to define a list of non-optimal configurations on the sample ASP.NET website. The report that Google PageSpeed generates includes suggestions for enabling browser caching, using compression, scaling images, and implementing bundling and minification. When you implement these suggestions, you'll see a reduction in the size of the page request and the time required to complete the rendering of the homepage.

You can choose any or all of the other online website analysis tools (such as YSlow and Modern.IE), correct the issues, and implement their suggestions. You might also consider some third-party solutions, such as ANTS Performance Profiler or the Performance Analyzer in Visual Studio. Both of these tools enable you to analyze ASP.NET performance, for example. Figure 3-2 illustrates the results of running Visual Studio Performance Analyzer on the ASP.NET website.

The report identifies the methods in the ASP.NET website that takes the longest to execute. This is useful information and you should use it to find places in your system that are not performing well and, therefore, should be optimized.

www.it-ebooks.info

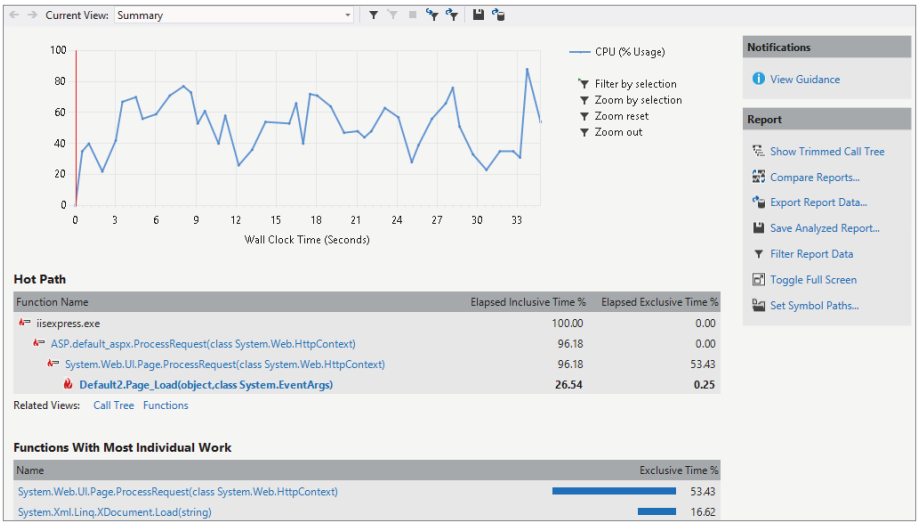


FIGURE 3-2

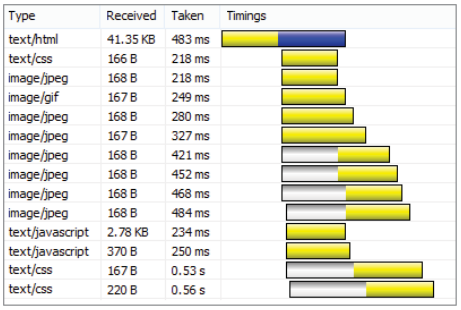
UNDERSTANDING BUNDLING AND MINIFICATION

Bundling and *minification* are features available in ASP.NET 4.5 and are introduced through the `System.Web.Optimization` namespace. They provide a mechanism for reducing the content size and number of round trips required to completely render a web page. The file types that generally benefit from either bundling or minification are typically JavaScript and CSS files, but are not limited to these types. For example, any file type that has a significant amount of white space or long variable names can benefit from this capability.

NOTE *Bundling and minification generally improve only the download speed of the first request. This is because the browser caches static content. Dynamic content is not a good candidate for bundling.*

Using the F12 Developer Tool Suite in Internet Explorer, you can see the total number of requests required to render a page, as well as the download speed and file sizes. Figure 3-3 shows three occurrences of `text/css` that have to a total size of 553 bytes and take 1.26 seconds to download. Likewise, two occurrences of `application/javascript` file types are requested to render a single page with a total size of 3.15 KB taking approximately 500 ms to download. As shown, each one of the files constitutes a separate `GET` request, and because there is a limit of six concurrent connections from a browser to the hostname, request number seven must wait until one of the other requests completes. The wait time is represented by the empty sections of the timing bar.

www.it-ebooks.info



Type	Received	Taken	Timings
text/html	41.35 KB	483 ms	
text/css	166 B	218 ms	
image/jpeg	168 B	218 ms	
image/gif	167 B	249 ms	
image/jpeg	168 B	280 ms	
image/jpeg	167 B	327 ms	
image/jpeg	168 B	421 ms	
image/jpeg	168 B	452 ms	
image/jpeg	168 B	468 ms	
image/jpeg	168 B	484 ms	
text/javascript	2.78 KB	234 ms	
text/javascript	370 B	250 ms	
text/css	167 B	0.53 s	
text/css	220 B	0.56 s	

FIGURE 3-3

If a web page on your website executes a lot of `GET` HTTP commands to render a page, you should certainly consider implementing bundling if not for the reason mentioned in the previous paragraph, then for the following reason: Requests traveling over a network should be compacted as much as possible. There is an overhead for network packet frames in the form of serialization, for example for each packet. Therefore, if you can reduce the number of frames, you can realize an additional performance gain. Figure 3-4 illustrates a chatty request (top figure) versus a chunky request (bottom figure). A chatty request is one which has a large number of frequent small packets, while a chunky request has a small number of large packets.

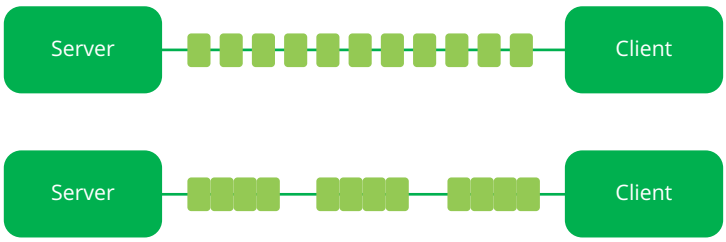


FIGURE 3-4

In summary, you can implement bundling with few lines of code, as discussed in Chapter 4. This reduces the number of `GET` requests that the client makes to the server by merging similar file types together into a single file. This also reduces any delay caused by connection limits and network packet framing.

> **NOTE** *Excessive bundling can increase latency and data loss, so test your solution thoroughly before going live with it.*

www.it-ebooks.info

The code shown in Listing 3-1 defines how to bundle a group of JavaScript files together. The requirements are to first create an instance of the `Bundle` class, which is part of the `System.Web.Optimization` namespace. You then identify the files and their relative path and add them to the `Bundles` object.

LISTING 3-1: Bundling JavaScript Files

```
void Application_Start(object sender, EventArgs e)
{
  Bundle JSBundle = new Bundle("~/JSBundle");
  JSBundle.Include("~/syntax/scripts/shCore.js");
  JSBundle.Include("~/syntax/scripts/shBrushCSharp.js");
  BundleTable.Bundles.Add(JSBundle);
}
```

Four constructors for the `Bundle` class are available:

- `Bundle(string virtualPath)`
- `Bundle(string virtualPath, params IBundleTransform[] transforms)`
- `Bundle(string virtualPath, string cdnPath)`
- `Bundle(string virtualPath, string cdnPath, params IBundleTransform[] transforms)`

If you review Figure 3-3, you'll notice the size of each individual file and the combined size per file type. Likewise, take note of the `params IBundleTransform[]) transforms` parameter of the `Bundle` class. Passing a class that implements the `IBundleTransform` interface results in the included files being minified. Therefore, implementing minification is as simple as passing an additional parameter to the `Bundle` class. Listing 3-2 is an example of the minification of the JavaScript files.

LISTING 3-2: Minifying JavaScript Files

```
void Application_Start(object sender, EventArgs e)
{
  Bundle JSBundle = new Bundle("~/JSBundle", new JsMinify());
  JSBundle.Include("~/syntax/scripts/shCore.js");
  JSBundle.Include("~/syntax/scripts/shBrushCSharp.js");
  BundleTable.Bundles.Add(JSBundle);
}
```

Notice the addition of the `new JsMinify()`, which is a default class found within the `System.Web.Optimization` namespace and which implements the `IBundleTransform` interface as required. When the minified CSS and JavaScript bundles are referenced from the requested pages, the requests are fewer and the response times faster, as shown in Figure 3-5.

www.it-ebooks.info

URL	Type	Received	Taken
http://mvc-4.cloudapp.net/Blogs/BlogLoca...	text/html	24.70 KB	343 ms
/CSSBundle?v=yeSI1SaGY6HIs-ImN629oF...	text/css	227 B	314 ms
/JSBundle?v=BhFExV_bSUI4ouA_67eGCJd...	text/javascript	2.72 KB	418 ms

FIGURE 3-5

This is a nice feature that is easy to implement, and for which gains in performance are quickly realized.

SCALING A WINDOWS AZURE CLOUD SERVICE

Before the cloud and virtual machines, scaling a system took a lot of time. Scaling meant that you needed to order a new server, to install the operating system and to connect to the network, to install the application onto it, and, if all went well, to add the server to the web farm and start directing traffic to it. This process could take weeks if not months from start to finish. Metrics and processes were required to forecast and anticipate growth and usage of the system well in advance so that you could scale out the current environment.

The creation of virtual machines changed that a lot. Tools, such as VMware and Hyper-V, speed things up considerably. To scale, you can take a snapshot of an existing virtual machine and use that snapshot to build another virtual instance of your system. It still takes time to get it added to the network and to load user traffic onto it, but nothing like it was before virtual machines.

Windows Azure scaling has taken it to the next level. While monitoring your system, if you notice that the CPU or memory usage increases, you can click the Scale link of the Cloud Service or Web Site and increase the number of instances, as shown in Figure 3-6.

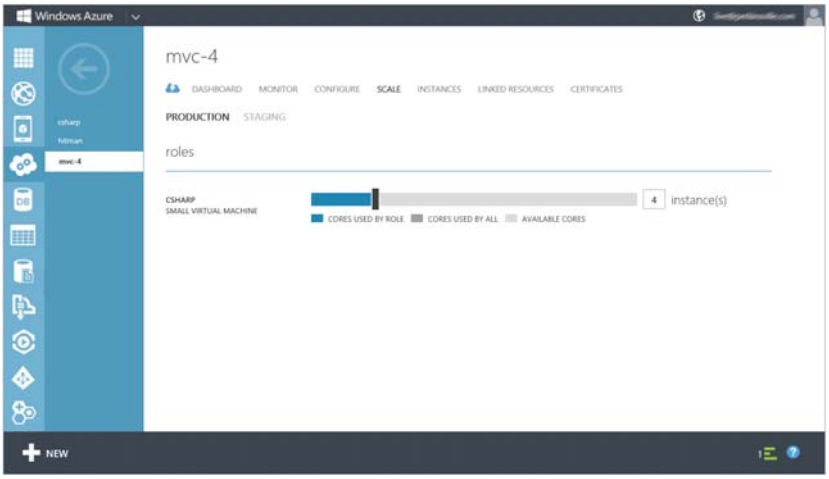


FIGURE 3-6

www.it-ebooks.info

In this case, the number of virtual machines running the `http://mvc-4.cloudapp.net` Web Role is increased to four. After you set and save the scaling of the Web Role, four virtual machines are created, the website is published to the virtual machines, and traffic begins flowing to the new instances when they are needed. During the deployment of the virtual machines, you can track the status by clicking the Instances link and then the environment you want to scale out (for example, either Production or Staging). Figure 3-7 illustrates the transition status.

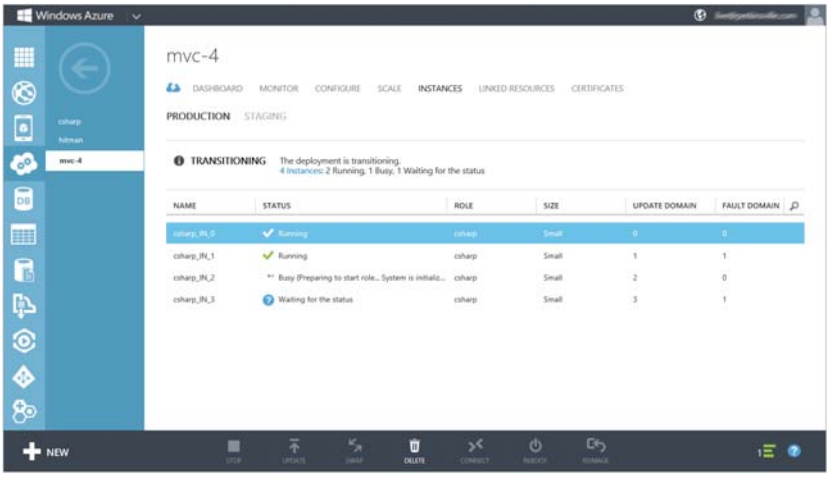


FIGURE 3-7

With three clicks, the website scales from one server to four. No manual configuration is required — it is all done for you automatically. When scaling Windows Azure Web Sites, you also need to take the Web Site Mode into account, as shown in Figure 3-8.

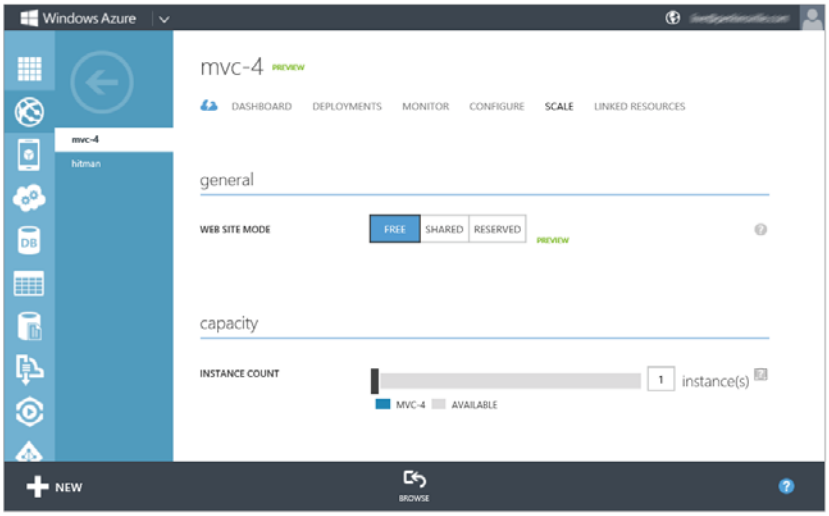


FIGURE 3-8

www.it-ebooks.info

The different website modes:

- **Free:** The constraints placed on this mode change, but the name of the mode says it all. Depending on your subscription, a number of websites with a bandwidth and storage limit are provided at no cost.
- **Shared:** As with Free mode, Shared mode is deployed into a multi-tenant environment. The main advantage is that there is no bandwidth limit.
- **Reserved:** Runs on an isolated small, medium or large virtual machine. There is no limit on the number of websites, CPU usage or memory.

When you click the "?" in the Web Site Mode section, the following information is rendered:

> *In the Free and Shared modes, all websites run in a multi-tenant environment and have quotas for usage of CPU, memory, and network resources. You can decide which sites you want to run in Free mode and which sites you want to run in Shared mode. Shared mode employs less stringent resource usage quotas than Free mode. The maximum number of sites you can run in Free mode may vary with your plan. When you choose Reserved mode, all your web sites run in Reserved mode on dedicated virtual machines that correspond to standard Windows Azure compute resources.*

This means that when your website is in Free or Shared mode, your site runs on a server with other websites. This is the common web hosting approach and is what the product Antares delivers to web-hosting companies. When you run your website in Reserved mode, it runs on a dedicated virtual machine similar to how a Web Role works.

Scaling is no longer a difficult nor costly endeavor. For example, you can scale out the website or Web Role when you are running a marketing campaign and then scale back when the traffic reduces. This *scaling back* technique is complicated to perform in older scaling models because after you pay for the hardware build it and direct traffic to it, it generally stays, as do the maintenance/fixed costs. Scaling back is a new concept that Windows Azure has made available. Because of this, the costs of unnecessary extra capacity can be significantly reduced.

FIFTEEN PERFORMANCE ENHANCING TIPS

Using online tools, bundling, minification, and scaling can all help your application perform faster and better under load. However, these tools and techniques may not resolve all the performance problems you may encounter in an ASP.NET website. With that in mind, you can implement the following 15 tips into your ASP.NET website to improve performance results:

1. **Check the efficiency of ORM-generated SQL queries, and the number of times they execute per transaction.** Use a SQL Profiler to confirm both prior to going live with the solution.

www.it-ebooks.info

2. **Turn on the compression of dynamic and static content that IIS supports.** Compressing the content results in smaller files, which, in turn, increases the speed of download.

3. **Don't log too much in your production environment.** You want your production systems to run as fast and as optimally as possible. Logging requires hardware resources that the application can use for doing its work. Although logging is necessary for troubleshooting, maintainability, and stability tracking, you'll want to log the minimum amount of information required and turn off your environment when it is not absolutely needed.

4. **Perform validations on the client prior to posting it to the server.** ASP.NET provides client-side validation controls. You can implement them and avoid an unnecessary round trip to the server. For example, consider the `<asp:RequiredFieldValidator />`. If you validate only on the server side, a post may fail and return an error message to the user. The user can then correct and then re-post.

5. **Use the `using{}` statement wherever possible and reduce memory leaks.** An example of a `using{}` statement follows:

```
using (SqlCommand command =
           new SqlCommand(sql,
           ConnectionManager.GetConnection()))
{
   command.CommandType = CommandType.Text;
   count = (int)command.ExecuteScalar();
}
```

   As soon as the code execution exits the `using{}` statement, the command object is marked for deletion and garbage collected the next time it runs.

6. **Set the `CacheControlMaxAge` attribute to a high number.** By setting this value to a higher number (for example, one year), on files that do not change (for example, HTML or image files on your website), you can ensure that requests are not retrieved unnecessarily. Only when the `CacheControlMaxAge` is exceeded will the file download again.

7. **Don't run your website in debug mode.** The web.config file for your website contains an attribute called debug located in the `<system.web><compilation debug="true">` tags. When in production, set this value to `false` to avoid unneeded overhead when you compile your ASP.NET temporary files.

8. **Although lazy loading is a good ORM technique, if you know the data will be retrieved later, select it, if possible, with a single database hit.** A feature called `Futures` in NHibernate enables you to bundle multiple database queries together. This means you can execute two or more queries and minimize the number of round trips to the database.

www.it-ebooks.info

9. **Always write the database query for the specific context.** Don't reuse the query if it selects data that is not needed. The more data you select, the longer the query takes to complete. In addition, the larger the amount of selected data stored in the result set, the longer it takes to travel back to the client and server across the network.
10. **Remove unused HTTP modules.** You can remove modules, such as `Profile`, `RoleManager`, `Session`, and `UrlAuthorizaion`, from the `globalModules` section if they are not used in your website. Even if they are not used, they consume memory.
11. **Use Asynchronous programming techniques.** One of the newer capabilities of ASP.NET is that long running code segments in a transaction can run on a separate thread. This enables code not dependent on the outcome of the long-running code segment to continue. After the long-running code segment completes, the threads merge back together.
12. **In MVC, use the `OutputCache`, which serves the request from memory instead of disk.** Accessing data in memory is much faster than accessing it on the hard drive. You should take every opportunity to store common result sets or reference data in memory.
13. **Implement paging at the database level, not on the client.** Paging means that you display a subset of the data in a `GridView`; then the user clicks a Next button to view the next page of data. The point is not to download thousands of rows of data and store them on the client for the user to page through them. It is likely the user will look at only the first few pages anyway. Therefore, implement the paging on the database side and select only the amount of data required per page.
14. **Hardware, hardware, hardware, and more hardware.** You can resolve many problems by increasing the capacity of the hardware. Because the price of hardware components is on the decline, you have no good reason not to have too much power.
15. **Disable `VIEWSTATE` if it is not needed.** By default, `VIEWSTATE` is enabled. You can disable it at the page level so that it is turned off for all controls or on a control-by-control basis. `VIEWSTATE` can get big, which increases the size of the file and impacts the download speed.

USEFUL LINKS

The following links can further help you optimize your system:

- **Tips for writing high-performance web applications:** `http://msdn.microsoft.com/en-us/magazine/cc163854.aspx`
- **The Miniprofiler official website:** `http://miniprofiler.com`
- **The ASP.NET checklist:** `http://webdevchecklist.com/asp.net/`

www.it-ebooks.info

- **Cost of performance:** You can find a good article that discusses the cost of performance on the Internet by searching for **Milliseconds are Money: How much performance matters in the Cloud.**
- **ASP.NET Caching:** `http://msdn.microsoft.com/en-us/library/aa478965.aspx`
- **Tool for testing your website in Internet Explorer:** `http://www.modern.ie`

SUMMARY

In this chapter, you learned the importance of performance in a website. Customers and visitors expect a website to be responsive and fast. If it does not meet their expectation, they will not return.

A number of tools can aid you in setting a baseline so that you can compare that baseline with performance measurements you take during a slow period. In addition, you can use tools such as ANTS and Visual Studio to gain deeper insight into which methods perform the slowest and, therefore, need more analysis and improvement.

You can implement features such as bundling, minification, and scaling to increase the performance and stability of your website in a short time. These features have a large positive impact on the website.

The next chapter has detailed exercises for implementing many of the concepts discussed in this chapter.

www.it-ebooks.info

4 Fine-tuning the ASP.NET MVC 4 Project for Performance

EXERCISES AND EXAMPLES

IN THIS CHAPTER

- Capturing performance statistics with Fiddler
- Using MiniProfiler to implement a website and Web Role
- Utilizing Internet Explorer F12 Developer tools to capture performance data
- Using Google PageSpeed on your ASP.NET website
- Reducing JavaScript and CSS files by bundling and minifying
- Setting up compression and caching
- Fine-tuning ASP.NET MVC 4 performance

WROX.COM CODE DOWNLOADS FOR THIS CHAPTER

You can find the wrox.com code downloads for this chapter at `www.wrox.com/go/azureaspmvcmigration` on the Download Code tab. It is recommended that you download the code and review it so that you have a good understanding of what is about to be discussed. The steps in this chapter cover baseline performance setting techniques, and ASP.NET Web Forms and ASP.NET MVC 4 optimization features.

As discussed in the previous chapter, the performance of your system is critical to its success. If your website runs slow, visitors are very unlikely to return to it. This chapter has examples and tips on how to fine-tune and optimize the ASP.NET, ASP.NET MVC 4 website, and ASP.NET MVC 4 Web Role.

www.it-ebooks.info

Before you begin implementing any change, it is always a good idea to set a performance baseline for what you are about to change. By doing this, you can see how the change impacts your system. You may be surprised that a change you expected to have a big impact actually had no or even a negative impact. However, without knowing the current status, there is actually no way of telling how the change affected the system…unless it is absolutely obvious.

NOTE *The following exercises provide step-by-step instructions for optimizing the performance of your ASP.NET MVC 4 project. You can use the projects created in Chapter 2 as a starting point for this chapter's exercises. If you are not familiar with the concepts behind this chapter, please read through Chapter 3 before continuing.*

USING FIDDLER TO CAPTURE PERFORMANCE STATISTICS

To capture some performance statistics, you can use a free tool called Fiddler. Although Fiddler is commonly used to troubleshoot issues happening between the browser and the web server, you can also use it to see how fast a web page can render. For this section, the purpose of these statistics is to set a baseline for you to use when you compare the results of optimizations performed later in this chapter.

To use Fiddler to capture performance statistics for both the ASP.NET website and ASP.NET MVC 4 project, perform the following steps:

1. Download and install the free Fiddler tool at `http://www.fiddler2.com`.
2. Open Fiddler and confirm the Capture Traffic menu item is checked, as shown in Figure 4-1.

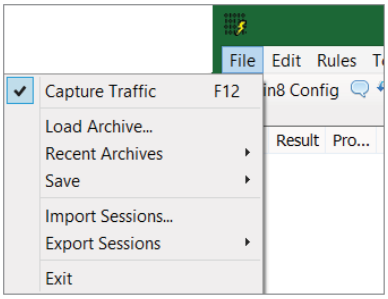


FIGURE 4-1

3. Open a web browser and access the web page you wish to measure using Fiddler. For example, `http://mvc-4.cloudapp.net/Blogs/BlogLocator/100`

www.it-ebooks.info

4. Highlight the requests in the Web Sessions panel. The top figure in Figure 4-2 shows the ASP.NET website (`http://aspnet.thebestcsharpprogrammerintheworld.com/blogs/Using-the-as-keyword-versus-boxing.aspx`), and bottom figure shows the ASP.NET MVC 4 (`http://mvc-4.cloudapp.net/Blogs/BlogLocator/100`).

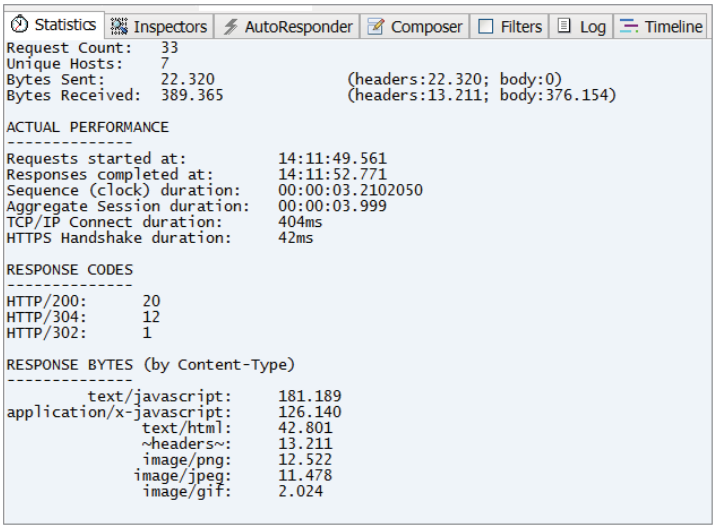
#	Result	Pro...	Host	URL	Body	Content-Type
1	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/blogs/Using-the-as-keyword-versus-boxing.aspx	42.120	text/html; charset=utf-8
2	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/include/style.css	2.660	text/css
3	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/newsletter.JPG	1.107	image/jpeg
4	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/RSS.gif	1.083	image/gif
5	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/home_t.JPG	1.846	image/jpeg
6	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/blogs_t_dark.JPG	1.902	image/jpeg
7	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/lessons_t.JPG	2.185	image/jpeg
8	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/reviews_t.JPG	2.111	image/jpeg
9	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/news_t.JPG	1.872	image/jpeg
10	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/help_t.JPG	1.749	image/jpeg
13	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/c-sharp.JPG	4.063	image/jpeg
15	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/WebResource.axd?d=s8Xjaa5NSzUkN9U0lirEC...	20.794	application/x-javascript
16	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/ScriptResource.axd?d=ngKIZxnNA6gdWLjgEW...	21.615	application/x-javascript; ...
17	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/ScriptResource.axd?d=I1xGDrpGCe5SivPiqraT...	99.445	application/x-javascript; ...
18	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/ScriptResource.axd?d=s2IEAca-fCST64kbT7An...	32.226	application/x-javascript; ...
19	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/ScriptResource.axd?d=kDDw2nkCs5K26ymmx...	27.531	text/javascript; charset=...
20	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/ScriptResource.axd?d=o3LRm7q9Bzroed0fspO...	20.300	text/javascript; charset=...
21	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/syntax/scripts/shCore.js	16.175	application/x-javascript
22	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/syntax/scripts/shBrushCSharp.js	2.528	application/x-javascript
23	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/syntax/styles/shCoreFadeToGrey.css	8.739	text/css
25	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/NHibernate_small.JPG	2.809	image/jpeg
26	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/IIS8.jpg	49.257	image/jpeg
29	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/blogs/images/5.PNG	3.700	image/png
30	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/blogs/images/6.PNG	4.031	image/png
31	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/blogs/images/7.PNG	3.733	image/png
33	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/titlebar.JPG	5.733	image/jpeg
36	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/titlebar_sub_nav.JPG	960	image/jpeg
37	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/eyebrow.JPG	803	image/jpeg
42	200	HTTP	aspnet.thebestcsharpprogrammerintheworld.com	/images/FilledStar.png	330	image/png

#	Result	Pro...	Host	URL	Body	Content-Type
1	200	HTTP	mvc-4.doudapp.net	/Blogs/BlogLocator/100	25.039	text/html; charset...
2	200	HTTP	mvc-4.doudapp.net	/Content/css?v=-Nh-8YI-4Neki6MMdTWZqai...	7.818	text/css; charset=u...
3	200	HTTP	mvc-4.doudapp.net	/bundles/modernizr?v=jmdBhqkI3eMaPZJdu...	10.875	text/javascript; cha...
4	200	HTTP	mvc-4.doudapp.net	/Content/syntax/scripts/shCore.js	16.175	application/javascript
5	200	HTTP	mvc-4.doudapp.net	/Content/syntax/scripts/shBrushCSharp.js	2.528	application/javascript
6	200	HTTP	mvc-4.doudapp.net	/Content/syntax/styles/shCoreFadeToGrey.css	8.739	text/css
9	200	HTTP	mvc-4.doudapp.net	/Images/NHibernateSmall.jpg	15.870	image/jpeg
10	200	HTTP	mvc-4.doudapp.net	/Images/iis8.JPG	49.257	image/jpeg
11	200	HTTP	mvc-4.doudapp.net	/Images/5.PNG	3.700	image/png
12	200	HTTP	mvc-4.doudapp.net	/Images/6.PNG	4.031	image/png
13	200	HTTP	mvc-4.doudapp.net	/Images/7.PNG	3.733	image/png
15	200	HTTP	mvc-4.doudapp.net	/bundles/jquery?v=wBUqTIMTmGl9Hj0haQM...	93.229	text/javascript; cha...
17	200	HTTP	mvc-4.doudapp.net	/Images/heroAccent.png	461	image/png

FIGURE 4-2

5. On the right side of the Fiddler console, you see a group of tabs. Select the Statistics tab to see results similar to Figure 4-3. The top figure shows the ASP.NET website and the bottom shows the ASP.NET MVC 4 Web Role.

www.it-ebooks.info



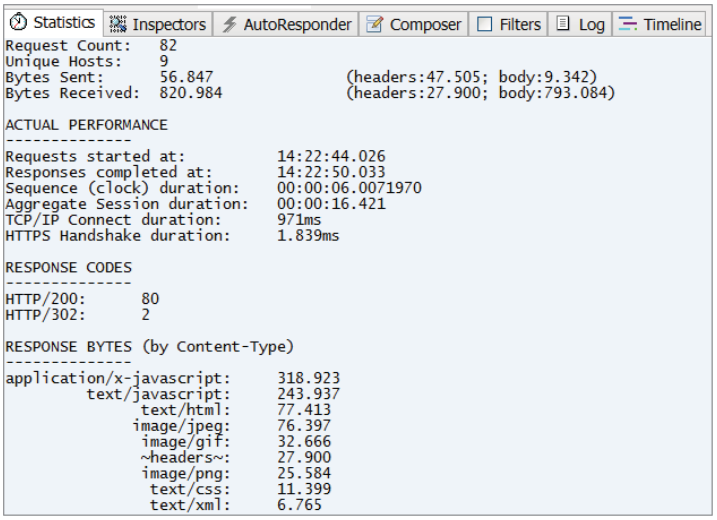


FIGURE 4-3

6. Using the data logged in the Statistics tab, you can get a good view of the size and performance of the page. Also, if you perform a search from any Internet search engine for **web page analyzer,** you can find some links to other tools that perform similar types of analyses online. Choose one and build a baseline for your system or for this exercise.

www.it-ebooks.info

For the following exercises, the statistics captured for baseline analysis and the basis for improvement are shown in Table 4-1 (for the ASP.NET Website) and Table 4-2 (for the ASP.NET MVC 4 Web Role).

NOTE *As shown in Table 4-1 and 4-2, some of the Values have a range. A number of tools were used to capture and view the performance of the webpage instead of relying solely on a single tool.*

TABLE 4-1: ASP.NET website Performance Baseline

ATTRIBUTE	VALUE
# of HTTP requests (objects)	29–61
Total size (bytes)	205 K–745.9 KB
Total download time (seconds)	3.03–6.89
HTML size (bytes)	36.7–79.2
HTML download time (seconds)	.39
CSS size (bytes)	11.4
CSS download (seconds)	.46
Image size (bytes)	94.2–104.8
Image download (seconds)	4.3
Script size (bytes)	63.3–547.2
Script download (seconds)	1.74
Total header sizes (bytes)	23.6 KB

TABLE 4-2: ASP.NET MVC 4 Web Role Baseline

ATTRIBUTE	VALUE
# of HTTP requests (objects)	14–33
Total size (bytes)	161 KB–376 KB
Total download time (seconds)	3.2–3.7
HTML size (bytes)	5.5–42.8
HTML download time (seconds)	1.67

www.it-ebooks.info

TABLE 4-2: *(Continued)*

ATTRIBUTE	VALUE
CSS size (bytes)	1.9 KB
CSS download (seconds)	.21
Image size (bytes)	25.9KB–88 KB
Image download (seconds)	1.67
Script size (bytes)	65.9–307.3 KB
Script download (seconds)	1.55
Total header sizes (bytes)	13.2

The statistics captured for the attributes in Table 4-1 and Table 4-2 are from the following URLs:

- **ASP.NET website:** `http://aspnet.thebestcsharpprogrammerintheworld.com/blogs/Using-the-as-keyword-versus-boxing.aspx`
- **ASP.NET MVC 4 Web Role:** `http://mvc-4.cloudapp.net/Blogs/BlogLocator/100`

IMPLEMENTING MINIPROFILER

Before tuning the website and Web Role, you can implement a tool called MiniProfiler. MiniProfiler is a single .NET assembly that, when configured, displays the timing of specific code segments in an ASP.NET page. For example, you can wrap it around a method or a specific line of code in a method. You can also wrap it around an entire ASP.NET page from start to finish.

NOTE *You can find more information about the MiniProfiler at* `http://miniprofiler.com/`.

The following exercises in this section provide the implementation instructions for both the sample ASP.NET website and sample ASP.NET MVC 4 project. The MiniProfiler is imbedded to track the performance of the following features contained in both the ASP.NET website and ASP.NET MVC 4 Web Role samples:

- Total time to load the homepage
- Time required to load the RSS XML data feed on the homepage
- Total time to load the sample blog provided in the example
- Time taken to load the Rating (ASP.NET website-only)

www.it-ebooks.info

- Time taken to load the blog comments
- Time taken to load each of the dynamic data content on the `BlogNavBar`

Implementing into ASP.NET Website

You can begin by implementing the MiniProfiler capabilities into the ASP.NET website project.

1. Download the ASP.NET website sample code, and open the CSHARP.ASP.NET website project in Visual Studio 2012.
2. Download the MiniProfiler component from `http://miniprofiler.com/` or perform step 3.
3. In Visual Studio select Tools ➪ Library Package Manager ➪ Package Manager Console, and the console opens, as shown in Figure 4-4.

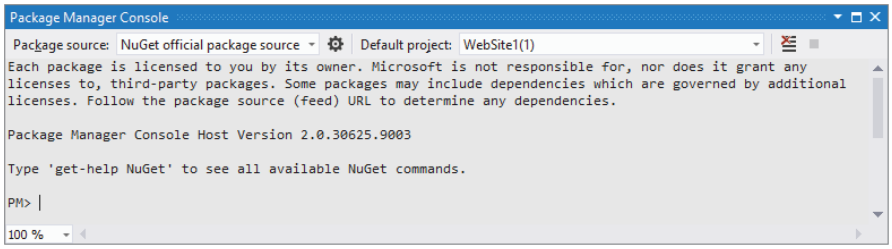

Package Manager Console
Package source: NuGet official package source
Default project: WebSite1(1)
Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages. Some packages may include dependencies which are governed by additional licenses. Follow the package source (feed) URL to determine any dependencies.
Package Manager Console Host Version 2.0.30625.9003
Type 'get-help NuGet' to see all available NuGet commands.
PM>
100 %

FIGURE 4-4

4. To install the MiniProfiler, enter the `install – package MiniProfiler` command, as shown in Figure 4-5.

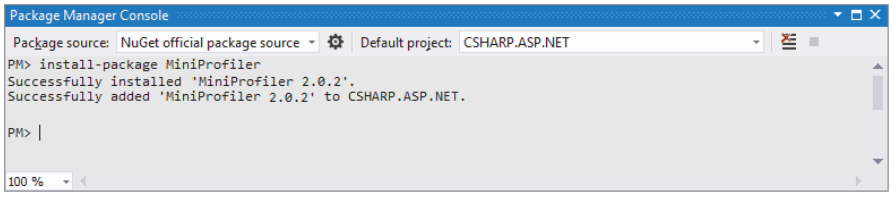


FIGURE 4-5

5. The ASP.NET website does not currently have a Global.asax file, but the configuration of the MiniProfiler is best when one exists; therefore, add one to the project. This is accomplished by right-clicking the Csharp.asp.NET project and then clicking Add ➪ Add New Item ➪ Global Application Class. Finally, click the Add button, leaving the name as Global.asax. Figure 4-6 illustrates this in more detail.

www.it-ebooks.info

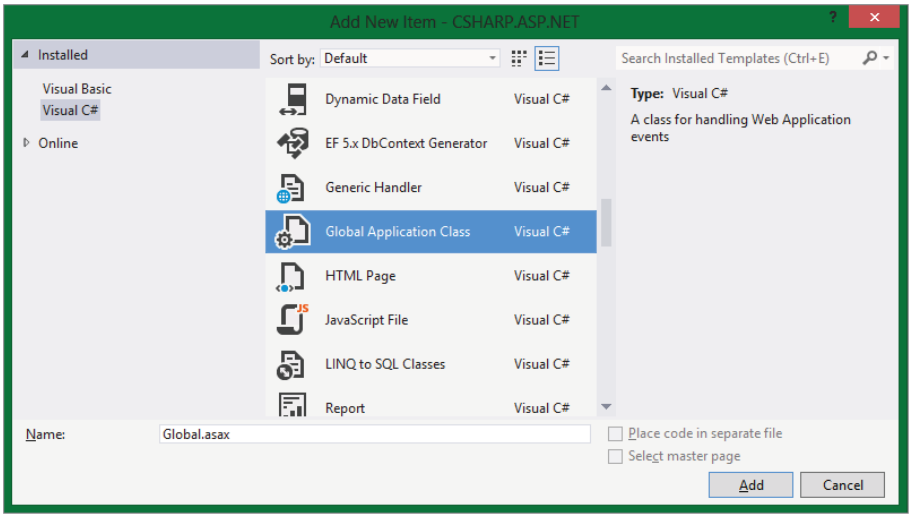


FIGURE 4-6

Adding the Global Class to the App_Code Directory

You will certainly notice that there is no code-behind when you add the file from the previous section. Historically, a Global.asax file is not needed in an ASP.NET Website project, and therefore they were/are not added by default. Also, Microsoft moved away from a code-behind file in this context in favor of the script-based code you see when the Global.asax file is opened. For this example, you simulate a code-behind for the Global.asax by adding a new class named Global to the App_Code directory.

To do this, follow these steps:

1. Remove all the code from the Global.asax file excluding the following code snippet:

```
<%@ Application Inherits="Global" Language="C#" %>
```

2. Right-click the `App_Code` directory in the Csharp.ASP.NET Website project and select Add ➪ Class. Then enter **Global** and click OK, as shown in Figure 4-7.

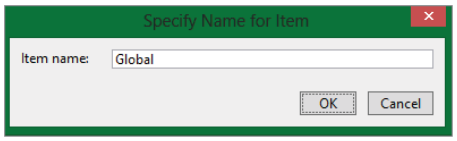


FIGURE 4-7

3. Open the `App_Code\Global.cs` class, and make the changes so it contains what is shown in Listing 4-1.

www.it-ebooks.info

NOTE *Listing 4-1 contains the configurations required by the MiniProfiler. The code listing begins tracing at the beginning of the request until the end. This is what delivers the time taken for rendering the homepage and the example blog. Therefore, no specific-code addition is required for those two files.*

LISTING 4-1: The Simulated Global.asax.cs Code-behind Class with MiniProfiler Configurations

```
using System;
using StackExchange.Profiling;

public class Global : System.Web.HttpApplication
{
    protected void Application_BeginRequest()
    {
        MiniProfiler profiler = null;
        profiler = MiniProfiler.Start(ProfileLevel.Verbose);
    }

    protected void Application_EndRequest()
    {
        MiniProfiler.Stop(true);
    }
}
```

Configuring and Loading MiniProfiler

Now that you have the code-behind sorted out, you can begin to configure the MiniProfiler and then load it. This includes determining the time it takes to load the homepage, sample blog, rating, blog comments, and dynamic data.

You can do so by following these steps:

1. In the web.config file located in the root directory of the ASP.NET website, add the code shown in listing 4-2 just before the closing configuration tag, that is, `</configuration>`. If the `<system.webServer>` or `<handlers>` tags already exist, do not duplicate them — add only what is missing.

LISTING 4-2: MiniProfiler Configuration — web.config

```
  <system.webServer>
    <handlers>
      <add name="MiniProfiler" path="mini-profiler-resources/*"
           verb="*" type="System.Web.Routing.UrlRoutingModule"
```


www.it-ebooks.info

LISTING 4-2: *(Continued)*

```
            resourceType="Unspecified" preCondition="integratedMode" />
    </handlers>
  </system.webServer>
```

2. Open the `TopLevel.master` file, and add the following code snippet just before the `</head>` tag. Do the same to the `SecondLevel.master` page:

```
<%= StackExchange.Profiling.MiniProfiler.RenderIncludes() %>
```

3. Right-click the project, and select Use IIS Express, as shown in Figure 4-8.

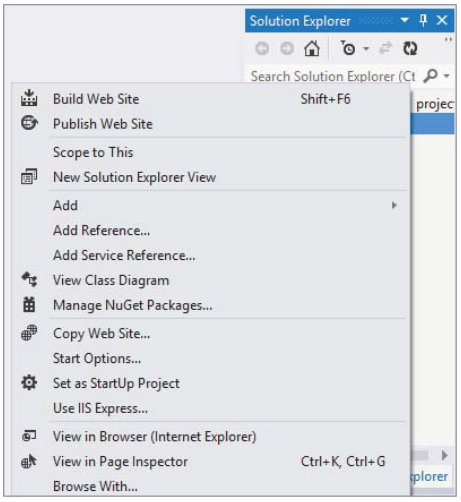


FIGURE 4-8

4. If the IIS Express menu option is not available, close down the solution, and reopen the website using File ➪ Open ➪ Web Site, and select the Csharp.ASP.NET website.
5. Click the Open button, and you receive a prompt asking if you want to use Cassini or IIS Express. Choose IIS Express.
6. Open the `Default.aspx.cs` file and make the following changes to the `Page_Load()` method, as shown in Listing 4-3.

LISTING 4-3: RSS XML Load — MiniProfiler

```
using StackExchange.Profiling;
protected void Page_Load(object sender, EventArgs e)
```


www.it-ebooks.info

```
{
  var mp = MiniProfiler.Current;
  using (mp.Step("RSS XML Load "))
  {
     LoadBlogXML();
  }
}
```

7. Open the `Blogs\Using-the-as-keyword-versus-boxing.aspx` file, and then find and modify the code segment that loads the Rating control, as shown in Listing 4-4.

LISTING 4-4: Blog Rating — MiniProfiler

```
<%@ Import Namespace="StackExchange.Profiling" %>

<asp:TableRow><asp:TableCell>
  <% using (MiniProfiler.Current.Step("Blog Rating Load"))
  {%>
      <RATING:Rating runat="server" ID="RatingControl" />
  <% } %>
</asp:TableCell></asp:TableRow>
```

8. Implement the profiler around the loading of the blog comments by adding the code, as shown in Listing 4-5.

LISTING 4-5: Blog Comment — MiniProfiler

```
<asp:TableRow><asp:TableCell>
  <% using (MiniProfiler.Current.Step("Blog Comments Load"))
  {%>
       <FEEDBACK:fbFORM runat="server" ID="FEEDbackForm" />
   <% } %>
</asp:TableCell></asp:TableRow>
```

9. The last file to modify in this exercise is the `Include\BlogRightColumn.ascx.cs`. In this file four methods are profiled:

   - Total number of Blogs ➪ `DisplayTotalBlogs()`
   - Total number of Comments ➪ `DisplayTotalBlogComments()`
   - Total number of Fundamentals ➪ `DisplayTotalFundamentalComments()`
   - Blog Archive list ➪ `DisplayBlogArchive()`

   Add the code shown in Listing 4-6.

www.it-ebooks.info

LISTING 4-6: Total # of Blogs - MiniProfiler

```
using StackExchange.Profiling;
protected int DisplayTotalBlogs()
{
  var mp = MiniProfiler.Current;
  using (mp.Step("DisplayTotalBlogs"))
  {
    return PostsDAC.PostTotalCount();
  }
}
```

10. Add the same code pattern as shown in Listing 4-6 to the remaining three methods referenced in the previous four items presented in the bulleted list.

Running the ASP.NET Website

Now that you have all the pieces together, you can run your site, see how it performs, and migrate everything to the server. To do so, follow these steps:

1. Run the ASP.NET website within Visual Studio 2012 by pressing F5, and you see the MiniProfiler tab, as shown in Figure 4-9.

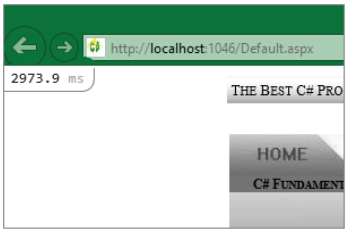


FIGURE 4-9

2. Click the tab and view the results. Be sure to also navigate to the sample blog to view the response times of the dynamic content.
3. Migrate your modifications to the server hosting the ASP.NET website, and document the baseline results, as shown in Table 4-3. You need to migrate the ASP.NET website to the server where you want to realize the performance gain. Steps 1 and 2 are only necessary to confirm that the website runs OK and that the performance isn't worse than expected.

TABLE 4-3: MiniProfiler ASP.NET Website Baseline

PROFILED FEATURE	TIME TAKEN (MS)
Homepage	2945.4
RSS XML load	8.9

www.it-ebooks.info

Sample Blog page load	4372.8
Blog Rating load	18.3
DisplayTotalBlogs	181.2
DisplayTotalBlogComments	883.8
DisplayTotalFundamentals	2639.7
DisplayBlogArchive	231.6

The following are the modified files that you need to publish so that they can measure the performance on a server and not a PC:

- Bin\MiniProfiler
- Global.asax
- App_Code\Global.cs
- web.config
- Default.aspx.cs
- TopLevel.master
- SecondLevel.master
- Blogs\Using-the-as-keyword-versus-boxing.aspx
- Include\BlogRightColumn.ascx.cs

Implementing into the ASP.NET MVC 4 Web Role

Now that the MiniProfiler is implemented in the ASP.NET website and the baseline timings are logged, you can implement the same into the ASP.NET MVC 4 Web Role.

Installing MiniProfiler

You need to install MiniProfiler, as shown in the following steps:

1. Downloaded the sample ASP.NET MVC 4 Web Role (`Chapter 4 – WindowsAzureMVC.zip`).
2. Unzip/extract the solution and open it in Visual Studio 2012.
3. Install the MiniProfiler using the Package Manager Console by selecting Tools ➪ Library Package Manager ➪ Package Manager Console and enter in the following command:

   ```
   Install-package MiniProfiler.MVC3
   ```

You should see a result similar to Figure 4-5, shown previously, while performing this command on the sample ASP.NET website.

www.it-ebooks.info

Implementing the Code to Track Features

When complete, begin to implement the code into the ASP.NET MVC 4 Web Role that configures and tracks timings of the chosen features, shown in the following list:

- Total time to load the homepage
- Time required to load the RSS XML data feed on the homepage
- Total time to load the sample blog provided in the example
- Time taken to load the blog comments
- Time taken to load each of the dynamic data content on the BlogNavBar

To do this, follow these steps.

1. Open the `_Layout.cshtml` file located in the `Views\Shared` directory of the project.
2. Add the following code shown snippet to the top of the page. Notice that a sample `_Layout.cshtml` file was added to the project named `_MINIPROFILER UPDATED_Layout.cshtml`. This file contains some examples of the configuration and components to profile.

```
@using StackExchange.Profiling;
```

3. Add the code shown in Listing 4-7 to the same `Views\Shared\_Layout.cshtml` file.

LISTING 4-7: Adding the RenderIncludes() Method — MiniProfiler

```
        @MiniProfiler.RenderIncludes()
    </body>
</html>
```

The `RenderIncludes()` method is placed at the end of the file just before the closing `</body>` tag.

4. Run the ASP.NET MVC 4 Web Role by pressing F5, and you see the profiler tab in the upper-left side of the web page, as shown in Figure 4-10.

FIGURE 4-10

www.it-ebooks.info

NOTE *The inclusion of the MiniProfiler into* `Views\Shared\_Layout.cshtml` *file tracks the time from the beginning of the request to the end for every page in the solution. Therefore, the homepage and sample blog do not need any additional configuration for the request time to be profiled.*

5. Open the `HomeController.cs` located in the `Controllers` folder, and modify the `Index()` method, as shown in Listing 4-8.

LISTING 4-8: RSS XML Load — MiniProfiler

```
using StackExchange.Profiling;

public ActionResult Index()
{
  ViewBag.Message = "This site is … Windows Azure.";
  var mp = MiniProfiler.Current;
  using (mp.Step("RSS XML Load"))
  {
    var blogs = BlogListXML();
    return View(blogs);
  }
}
```

6. Add the same code pattern to the `GetCommentsList(int Id)` method found in the `Controllers\BlogController.cs` file, as shown in Listing 4-9.

LISTING 4-9: DisplayTotalBlogComments — MiniProfiler

```
using StackExchange.Profiling;
public IList<Comments> GetCommentsList(int Id)
{
  var mp = MiniProfiler.Current;
  using (mp.Step("DisplayTotalBlogComments"))
  {
    using (ISession session = MvcApplication.SessionFactory.OpenSession())
    using (ITransaction transaction = session.BeginTransaction())
    {
      Blog blog = session.Get<Blog>(Id);
      IList<Comments> comments = blog.comments.ToList<Comments>();
      return comments;
    }
  }
}
```


www.it-ebooks.info

7. Capture the dynamic data presented in the `BlogNavBar` by opening the `Models\BlogNavBarElements.cs` file and wrap the queries in the MiniProfiler code, as shown in Listing 4-10. Listing 4-10 shows only the wrapping of the query that retrieves the total number of blogs.
8. Wrap the other three queries, which return the total number of comments, total number of fundamentals, and the blog archive list.

LISTING 4-10: DisplayTotalBlogs — MiniProfiler

```
using StackExchange.Profiling;

var mp = MiniProfiler.Current;
using (mp.Step("DisplayTotalBlogs "))
{
  using (ISession session = MvcApplication.SessionFactory.OpenSession())
  using (ITransaction transaction = session.BeginTransaction())
  {
    IQuery blogQuery = session.CreateQuery("Select count(*) from Blog");
    elements.blogCount = Convert.ToInt32(blogQuery.UniqueResult());
  }
}
```

After publishing the code changes to your Windows Azure Web Role, you can execute the request for the homepage and sample blog. Document the baseline timings so that you can compare them to the same timings after the optimizations. Table 4-4 represents the baseline timings logged by the profiler.

TABLE 4-4: MiniProfiler ASP.NET MVC 4 Web Role Baseline

PROFILED FEATURE	TIME TAKEN (MS)
Homepage	1351.1
RSS XML load	4.6
Sample Blog page load	2452.4
DisplayTotalBlogs	288.6
DisplayTotalBlogComments	34.9
DisplayTotalFundamentals	47.7
DisplayBlogArchive	35.9

Figure 4-11 illustrates an example of the MiniProfiler report after clicking the tab.

www.it-ebooks.info

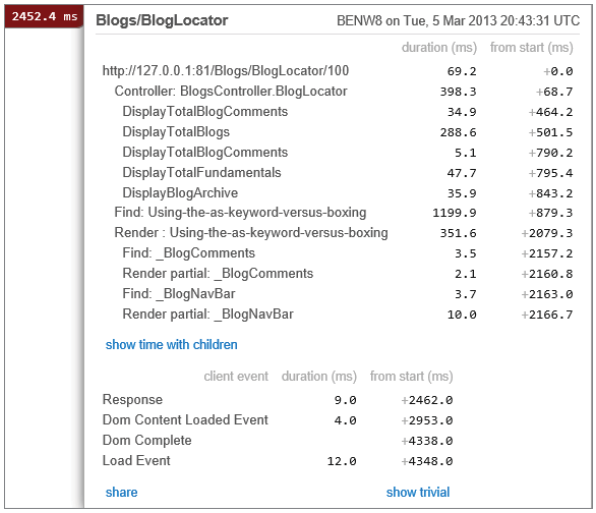


FIGURE 4-11

CAPTURING PERFORMANCE DATA WITH IE F12 DEVELOPER TOOLS

Another very useful tool for measuring performance and for debugging your website is the F12 Developer tool suite. To access these tools, follow these steps:

1. Open Internet Explorer and press F12.
2. Select the Network tab and then click the Start Capturing button.
3. Access the sample ASP.NET website at `http://aspnet.thebestcsharpprogrammerintheworld.com/blogs/Using-the-as-keyword-versus-boxing.aspx`. Press the Stop Capturing button. The result is a shown in Figure 4-12.

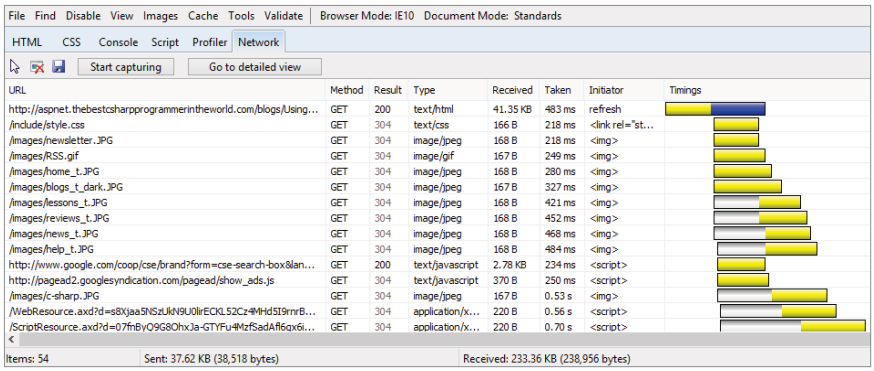

File Find Disable View Images Cache Tools Validate Browser Mode: IE10 Document Mode: Standards
HTML CSS Console Script Profiler Network
Start capturing Go to detailed view
URL Method Result Type Received Taken Initiator Timings
http://aspnet.thebestcsharpprogrammerintheworld.com/blogs/Using... GET 200 text/html 41.35 KB 483 ms refresh
/include/style.css GET 304 text/css 166 B 218 ms <link rel="st...
/images/newsletter.JPG GET 304 image/jpeg 168 B 218 ms <img>
/images/RSS.gif GET 304 image/gif 167 B 249 ms <img>
/images/home_t.JPG GET 304 image/jpeg 168 B 280 ms <img>
/images/blogs_t_dark.JPG GET 304 image/jpeg 167 B 327 ms <img>
/images/lessons_t.JPG GET 304 image/jpeg 168 B 421 ms <img>
/images/reviews_t.JPG GET 304 image/jpeg 168 B 452 ms <img>
/images/news_t.JPG GET 304 image/jpeg 168 B 468 ms <img>
/images/help_t.JPG GET 304 image/jpeg 168 B 484 ms <img>
http://www.google.com/coop/cse/brand?form=cse-search-box&lan... GET 200 text/javascript 2.78 KB 234 ms <script>
http://pagead2.googlesyndication.com/pagead/show_ads.js GET 304 text/javascript 370 B 250 ms <script>
/images/c-sharp.JPG GET 304 image/jpeg 167 B 0.53 s <img>
/WebResource.axd?d=s8Xjaa5NSzUkN9U0lrECKL52Cz4MHd5I9rnrB... GET 304 application/x... 220 B 0.56 s <script>
/ScriptResource.axd?d=07fnByO9G8OhxJa-GTYFu4MzfSadAfl6qx6i... GET 304 application/x... 220 B 0.70 s <script>
Items: 54
Sent: 37.62 KB (38,518 bytes)
Received: 233.36 KB (238,956 bytes)

FIGURE 4-12

www.it-ebooks.info

This is a very useful view that details all the requests made to the website, how large they are, and how long they took. The progress bar in the Timings column represents the request versus the response time. The yellow represents the time it takes for the request to occur, the blue represents the time it takes to get a response, and the gray represents the queue time.

4. Perform the same procedure previously discussed to set the baseline for the ASP.NET MVC 4 Web Role, located at `http://mvc-4.cloudapp.net/Blogs/BlogLocator/100`; the result is shown in Figure 4-13.

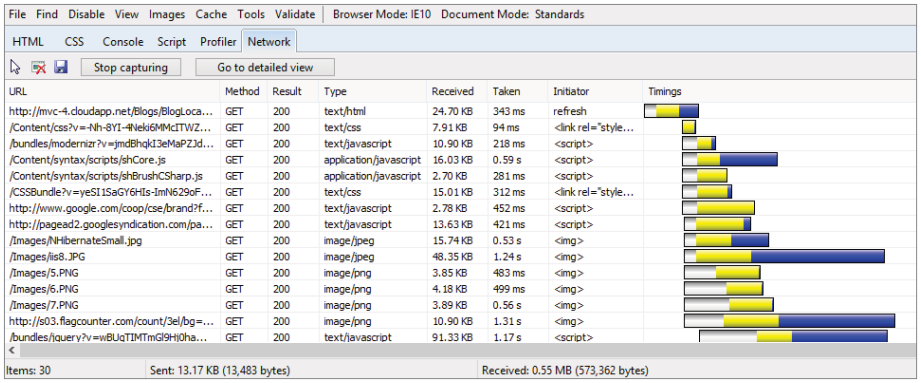

File Find Disable View Images Cache Tools Validate Browser Mode: IE10 Document Mode: Standards
HTML CSS Console Script Profiler Network
Stop capturing Go to detailed view
URL Method Result Type Received Taken Initiator Timings
http://mvc-4.cloudapp.net/Blogs/BlogLoca... GET 200 text/html 24.70 KB 343 ms refresh
/Content/css?v=-Nh-8Y1-4Neki6MMcITWZ... GET 200 text/css 7.91 KB 94 ms <link rel="style...
/bundles/modernizr?v=jmdBhqkI3eMaPZJd... GET 200 text/javascript 10.90 KB 218 ms <script>
/Content/syntax/scripts/shCore.js GET 200 application/javascript 16.03 KB 0.59 s <script>
/Content/syntax/scripts/shBrushCSharp.js GET 200 application/javascript 2.70 KB 281 ms <script>
/CSSBundle?v=yeSI1SaGY6HIs-ImN629oF... GET 200 text/css 15.01 KB 312 ms <link rel="style...
http://www.google.com/coop/cse/brand?f... GET 200 text/javascript 2.78 KB 452 ms <script>
http://pagead2.googlesyndication.com/pa... GET 200 text/javascript 13.63 KB 421 ms <script>
/Images/NHibernateSmall.jpg GET 200 image/jpeg 15.74 KB 0.53 s <img>
/Images/iis8.JPG GET 200 image/jpeg 48.35 KB 1.24 s <img>
/Images/5.PNG GET 200 image/png 3.85 KB 483 ms <img>
/Images/6.PNG GET 200 image/png 4.18 KB 499 ms <img>
/Images/7.PNG GET 200 image/png 3.89 KB 0.56 s <img>
http://s03.flagcounter.com/count/3el/bg=... GET 200 image/png 10.90 KB 1.31 s <img>
/bundles/jquery?v=wBUqTIMTmGl9HjOha... GET 200 text/javascript 91.33 KB 1.17 s <script>
Items: 30 Sent: 13.17 KB (13,483 bytes) Received: 0.55 MB (573,362 bytes)

FIGURE 4-13

If you perform these actions on one of your own websites or Web Roles, you might consider exporting the data by clicking the Save button. You can save the data as XML or CSV format, and use it later to compare it against a more fine-tuned or optimized ASP.NET project.

NOTE *The F12 Developer tool has many other useful features for developers and system administrators to better understand how a website is performing. The suite is also very useful for debugging JavaScript, jQuery, and other client-side website issues. Take a look at the numerous resources on the Internet for more details on this suite of tools.*

EMPLOYING GOOGLE PAGESPEED — ASP.NET WEBSITE

Google PageSpeed is a very powerful tool that performs analysis on a user-supplied web address. The result of the analyses is a score of somewhere between 0 and 100 — with 100 being the best. For example, entering the following URL into the tool results in a score of 78 points out of 100: `http://aspnet.thebestcsharpprogrammerintheworld.com/blogs/Using-the-as-keyword-versus-boxing.aspx`.

www.it-ebooks.info

In addition, one benefit of this online tool is that it provides numerous suggestions that can help the analyzed website perform more optimal. Also, PageSpeed renders a very valuable set of actions items that prioritize the reasons your site did not achieve the full 100 points, listing those actions having the most positive impact first. The actionable items provided after the analysis of the previously entered web address are provided in Table 4-5:

TABLE 4-5: Google PageSpeed Result

RECOMMENDATION	PRIORITY
Leverage Browser Caching	1
Enable Compression	2
Serve Scaled Images	2
Optimize Images	3
Bundle and Minify JavaScript, CSS, and HTML	3

When you click any of the suggestions, you navigate to a page describing the action required to implement the suggestion. The following sections illustrate how to implement the first four suggestions in Table 4-5 to see how it improves the score.

> **NOTE** *You can find Google PageSpeed at* `https://developers.google.com/speed/pagespeed/insights`, *or do an Internet search for Google PageSpeed to find the correct link.*

Leveraging Browser Caching

When you have *caching content* in the browser, it means that if the same file is requested again in the future, instead of performing an HTTP `GET` to retrieve the content, the content is loaded from the local cache. Because the ASP.NET sample website is held with a hosting provider, you have no possibility of configuring IIS to set the caching properties. In addition, this ASP.NET website is hosted on IIS 6, and there is no simple way to set the `Expires` header date for static content. Therefore, the configuration is performed in the web.config file in preparation for migration to IIS 7+. For example, the following configuration, shown in Listing 4-11, sets the expiration date of the static content to a date one year in the future.

LISTING 4-11: Setting the Expires Value in the web.config File

```
<system.webServer>
  <staticContent>
    <clientCache cacheControlMode="UseExpires"
```


www.it-ebooks.info

LISTING 4-11: *(Continued)*

```
                httpExpires="Mon, 03 Mar 2014 08:00:00 GMT" />
  </staticContent>
</system.webServer>
```

Alternatively, you can add the following code snippet at the top of the ASP.NET page. The snippet caches the page for approximately 30 days.

```
<%@ OutputCache Duration="25920000" VaryByParam="*" %>
```

Adding both of the elements just mentioned increases the Google PageSpeed by one point. Migrating the ASP.NET website to an IIS 7+ instance increases the score even more.

Enabling Compression

To enable compression using your web.config file add the following configuration, as shown in Listing 4-12.

LISTING 4-12: Configure Compression in Your web.config File

```
<system.webServer>
  <httpCompression
      directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files">
    <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" />
    <dynamicTypes>
      <add mimeType="text/*" enabled="true" />
      <add mimeType="message/*" enabled="true" />
      <add mimeType="application/javascript" enabled="true" />
      <add mimeType="*/*" enabled="false" />
    </dynamicTypes>
    <staticTypes>
      <add mimeType="text/*" enabled="true" />
      <add mimeType="message/*" enabled="true" />
      <add mimeType="application/javascript" enabled="true" />
      <add mimeType="*/*" enabled="false" />
    </staticTypes>
  </httpCompression>
  <urlCompression doStaticCompression="true" doDynamicCompression="true" />
</system.webServer>
```

NOTE *Both* `clientCache` *and* `staticCompression` *are accessible in the web.config only in IIS 7+. If these capabilities need to be implemented on an IIS 6 environment, it must be done with IIS.*

www.it-ebooks.info

After you add this configuration, the content types identified by the `mimeType` value are compressed. Adding and deploying these compression examples increases the PageSpeed score by 2 points.

Serve Scaled Images

There are two files identified in the report as being scaled incorrectly:

- IIS8.jpg is 49 KB and 421 × 529.
- c-sharp.jpg is 4 KB and 134 × 128.

It appears that these two images are larger than necessary, and when they are rendered, HTML or CSS resizes them to a smaller size. It makes sense to reduce the size of the files so that they match the size being rendered.

The IIS8.jpg is referenced in the Include\BlogRightColumn.ascx file and it does indeed include `Width="90"` and `Height="110"` values, which reduce its actual values from `width="421"` by `height="529"`. You must remove the `Width` and `Height` settings from the Include\BlogRightColumn.ascx and reduce the size of the image using an image editor, such as Microsoft Paint (mspaint.exe).

To reduce the image size, follow these simply steps:

1. Open the image in Paint.
2. Press CTRL+W to open the Resize and Show dialog.
3. Click the Pixels radio button.
4. Deselect the Maintain Aspect Ratio check box.
5. Enter **90** and **110** into the Horizontal and Vertical text boxes, respectively, as illustrated in Figure 4-14.
6. Make the same changes to the c-sharp.jpg and the SecondLevel.master page.

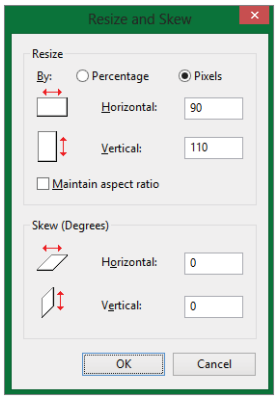


FIGURE 4-14

www.it-ebooks.info

The images are now smaller in size (IIS8.png is now 8 KB and c-sharp.png is now 4 KB) and no longer need to be reduced using HTML `width` and `height` attributes.

Optimizing Images

PNG image files are considered static files; however, they usually do not benefit from compression because these file are likely already compressed. Nonetheless, you can take some actions to reduce the size without losing the quality of the image.

The results of the Google PageSpeed analysis identified 16 image files, which may result in an approximate 17 KB reduction in the size of the files. This equates to a reduction of more than 25 percent in their original size. Using tools, such as PNGOUT or GIMP, to reduce the size and convert the images from JPG to PNG would create these gains.

NOTE *You can download PNGOUT from* `http://www.advsys.net/ken/util/pngout.htm` *and GIMP from* `http://www.gimp.org/`.

One of the images identified in the analysis report is the titlebar.jpg. By converting the image from a JPG to a PNG you may reduce the size by 45 percent. Opening the titlebar.jpg in Paint and performing a Save As ➪ PNG, results in a 33 percent reduction in the size of the file. Now see if you can get the 45 percent reduction using GIMP:

1. After GIMP is downloaded and installed, open the titlebar.jpg by selecting File ➪ Open.
2. Navigate to the location where the image is stored. Figure 4-15 illustrates this.

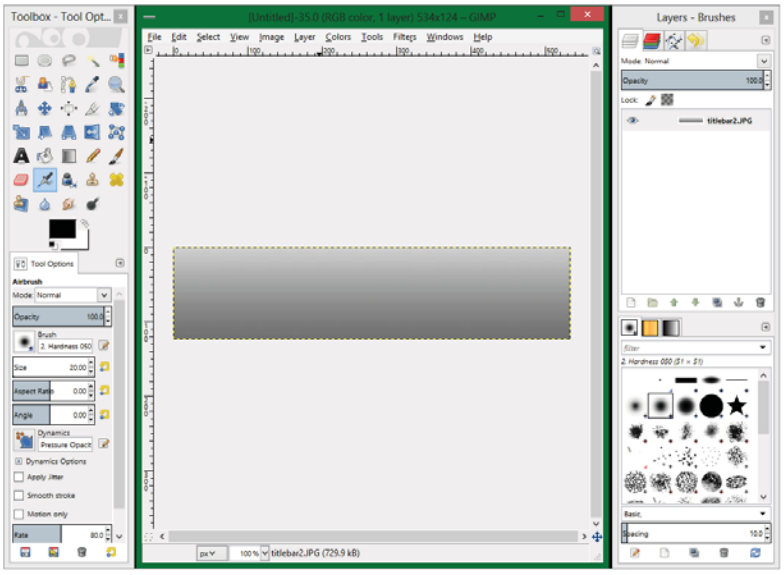


FIGURE 4-15

www.it-ebooks.info

3. Select File ➪ Export to open the Export Image dialog, as shown in Figure 4-16.

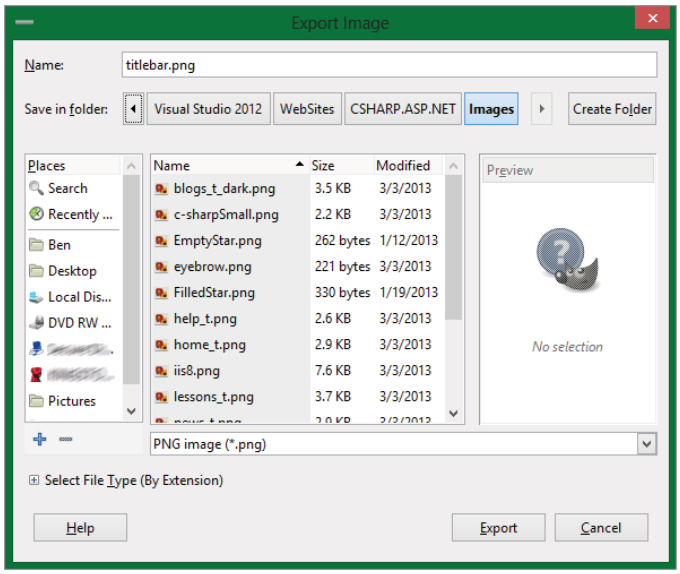


FIGURE 4-16

4. Enter **titlebar.png** in the Name field.
5. In the drop-down at the lower right select PNG Image (*.png).
6. Click the Export button, twice.
7. Leave the default settings in the second window.

When exported, check the size of the image. Instead of Paint's reduction of 33 percent, GIMP reduced the image size by 66 percent; this is even more than the original estimation.

If you perform the same method with the other JPGs and convert them to PNGs (see Table 4-6), you'll see that not every export does not result in an image size reduction.

TABLE 4-6: Image File Size Before and After Converting with GIMP

FILENAME	ORIGINAL SIZE	EXPORTED SIZE
Titlebar.jpg/png	6K B	2 KB
5.jpg/png	3.61 KB	3.49 KB
6.jpg/png	3.93 KB	3.87 KB
7.jpg/png	3.64 KB	3.60 KB

www.it-ebooks.info

TABLE 4-6: *(Continued)*

FILENAME	ORIGINAL SIZE	EXPORTED SIZE
IIS8.jpg/png	7 KB	15 KB
Lessons_t.jpg/png	3 KB	4 KB
Reviews_t.jpg/png	3 KB	4 KB
NHibernate_small.jpg/png	3 KB	7 KB
News_t.jpg/png	2 KB	3 KB
Home_t.jpg/png	2 KB	3 KB
Help_t.jpg/png	2 KB	3 KB
Blogs_t_dark.jpg/png	2 KB	4 KB
Newsletter.jpg/png	2 KB	1 KB
Titlebar_sub_nav.jpg/png	960 B	229 B
Eyebrow.jpg/png	803 B	221 B
C-sharp.jpg/png	2 KB	3 KB

You must modify the references made to an image when you notice a reduction in file size. You can do so by changing the image reference from .jpg to .png in the SecondLevel.master and the Blogs/Using-the-as-keyword-versus-boxing.aspx file.

NOTE *To further decrease image size, consider reducing the contrast, brightness, hue, saturation, and lightness so that the range of colors is minimized.*

NOTE *Check and see what Google Speed says about the ASP.NET MVC 4 Web Role. Navigate to the Google PageSpeed site and enter the following URL:* `http://mvc-4.cloudapp.net/Blogs/BlogLocator/100`.

BUNDLING AND MINIFYING JAVASCRIPT AND CSS

Bundling and minification are new to ASP.NET 4.5 and can improve performance by reducing the number of file downloads required to render a page and reducing the size of JavaScript, CSS, and HTML files. They're defined as:

www.it-ebooks.info

- **Bundling:** To combine a number of different files into a single file
- **Minification:** The removal of white space, comments, and the shortening of variable names

Understanding the Impact of Bundling and Minifying Files

To view the impact of implementing these features, follow these steps:

1. Using the F12 Developer tool, access the sample blog, for example `http://aspnet.thebestcsharpprogrammerintheworld.com/blogs/Using-the-as-keyword-versus-boxing.aspx`. Doing this can result in something similar to Figure 4-17.

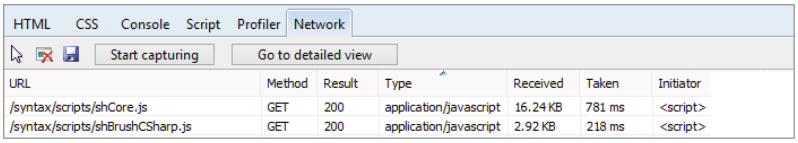


FIGURE 4-17

2. There are two GET requests for .js files and for .css files. This is a small website, but there are cases when a website would have 20 or more Cascading Style Sheets and 50 or more JavaScripts. Both file types can be bundled and minified, so the size is reduced and the number of requests to retrieve them is drastically reduced.

Implementing Bundling

To implement bundling into the ASP.NET website, perform the following steps:

1. Install the ASP.NET Web Optimization Framework into your project by opening the Package Manager.
2. Select Tools ➪ Library Package Manager ➪ Package Manager Console, and enter the following command:

   ```
   Install-package Microsoft.AspNet.Web.Optimization
   ```

3. Open the App_Code\Global.cs file, and add the following code shown in Listing 4-13.

LISTING 4-13: Bundling ASP.NET JavaScript Files

```
void Application_Start(object sender, EventArgs e)
{
  Bundle JSBundle = new Bundle("~/JSBundle");
  JSBundle.Include("~/syntax/scripts/shCore.js");
  JSBundle.Include("~/syntax/scripts/shBrushCSharp.js");
  BundleTable.Bundles.Add(JSBundle);
}
```


www.it-ebooks.info

4. Open the sample blog, Blogs\Using-the-as-keyword-versus-boxing.aspx, and add the following code snippet to the beginning of the file:

```
<%@ Import Namespace="System.Web.Optimization" %>
```

5. Comment out the current references to the JavaScript files, as shown in Listing 4-14, and add the new reference to the just-created JavaScript bundle shown in the following snippet:

```
//<script type="text/javascript"
        //src="<%= BundleTable.Bundles.ResolveBundleUrl("~/JSBundle") %>"></script>
```

LISTING 4-14: Individual JavaScript File References

```
<script type="text/javascript" src="../syntax/scripts/shCore.js"></script>
<script type="text/javascript" src="../syntax/scripts/shBrushCSharp.js"></script>
```

6. Run the ASP.NET website and access the sample blog. Using the F12 Developer Tool Suite, notice the two requests for the JavaScript files are no longer executed. As illustrated in Figure 4-18, there is only a single `GET` request for the `JSBundle` bundle.

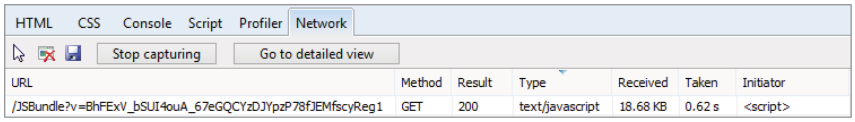


FIGURE 4-18

Implementing Minification

Compare the size and time taken between the bundled and nonbundled results (refer to Figures 4-17 and 4-18. Notice that the size reduced from 19.16 KB to 18.68 KB, and the time improved about 4/10th of a second. But you can get additional improvement after implementing minification.

To implement minification:

1. Open the `App_Code\Global.cs` file, and modify the `Application_Start()` method so that it resembles Listing 4-15.

LISTING 4-15: Add Minification to the Application_Start() Method

```
void Application_Start(object sender, EventArgs e)
{
  Bundle JSBundle = new Bundle("~/JSBundle", new JsMinify());
  JSBundle.Include("~/syntax/scripts/shCore.js");
```


www.it-ebooks.info

```
    JSBundle.Include("~/syntax/scripts/shBrushCSharp.js");
    BundleTable.Bundles.Add(JSBundle);
}
```

Notice that when you create the `Bundle`, the only addition is the `new JsMinify()` parameter.

2. Press F5 and run the ASP.NET website.
3. Use the IE F12 Developer Tools to see if there are additional improvements in size and speed. For an introduction to F12 Developer tools, you can find some information at this URL: `http://msdn.microsoft.com/en-us/library/ie/gg589512(v=vs.85).aspx`. Figure 4-19 provides the result of the test.

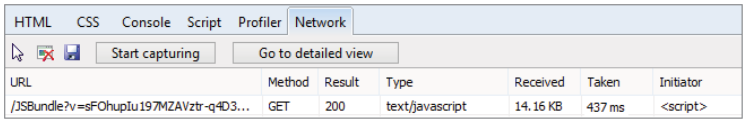


FIGURE 4-19

NOTE *Both the size and the time required to retrieve it decreased. This example is a simple one with only two JavaScript files. You might experience even better gains when you optimize more and/or larger files.*

4. Implement bundling and minification into the ASP.NET MVC 4 Web Role. After opening the sample ASP.NET MVC 4 Web Role in Visual Studio 2012, press F5 to run the website.
5. Navigate to the sample blog, accessing the sample blog results in two `GET` requests for the Cascading Style Sheets, as shown in Figure 4-20.

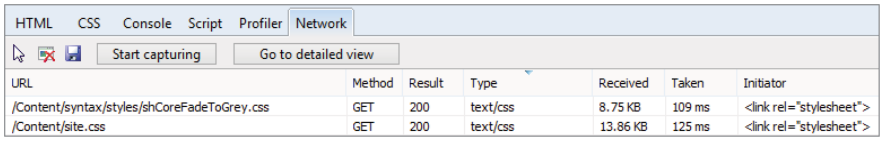


FIGURE 4-20

6. Create a Custom Bundle so that instead of two `GET` requests, only one is submitted and at the same time apply minification. To achieve this open the Global.asax.cs file and add the code shown in Listing 4-16 at the end of the existing `Application_Start()` method.

www.it-ebooks.info

LISTING 4-16: Bundling and Minification of CSS in ASP.NET MVC 4 Web Role

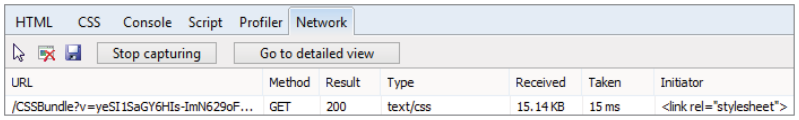
```
Bundle CSSBundle = new Bundle("~/CSSBundle", new CssMinify());
CSSBundle.Include("~/Content/Site.css");
CSSBundle.Include("~/Content/Syntax/Styles/shCoreFadeToGrey.css");
BundleTable.Bundles.Add(CSSBundle);
```

7. Open the `Views\Blogs\Using-the-as-keyword-versus-boxing.cshtml` and replace the existing `<link>` reference to the following code snippet:

```
<link type="text/css" rel="stylesheet"
      href="@BundleTable.Bundles.ResolveBundleUrl("~/CSSBundle")" />
```

   That's it! The Cascading Style Sheets for this page have been bundled and minified.

8. To check the impact, run the ASP.NET MVC 4 Web Role and check the size and time changes for the CSS files. Figure 4-21 represents the changes.



As you'll notice, there were some significant improvements for both size and speed in this example. Prior to implementing the bundling and minification the total time for download was 134 ms with a combined size of about 22 KB. After the implementation, the download time is 15 ms with a size of 15 KB. That's almost a 90 percent increase in performance.

CONFIGURING COMPRESSION AND CACHING

A benefit of utilizing a Windows Azure Web Role compared to a Windows Azure Web Site is making a Remote Desktop Connection to the server. This provides the administrator the ability to directly optimize configurations from the IIS Management console.

Implementing Compression

To create a Remote Desktop Connection and connect to the Windows Azure Web Role, review the exercises in Chapter 8 and then follow these steps to implement compression and caching:

1. When connected, open the IIS manager, and in the Features View, select the Compression icon, as shown in Figure 4-22.

www.it-ebooks.info

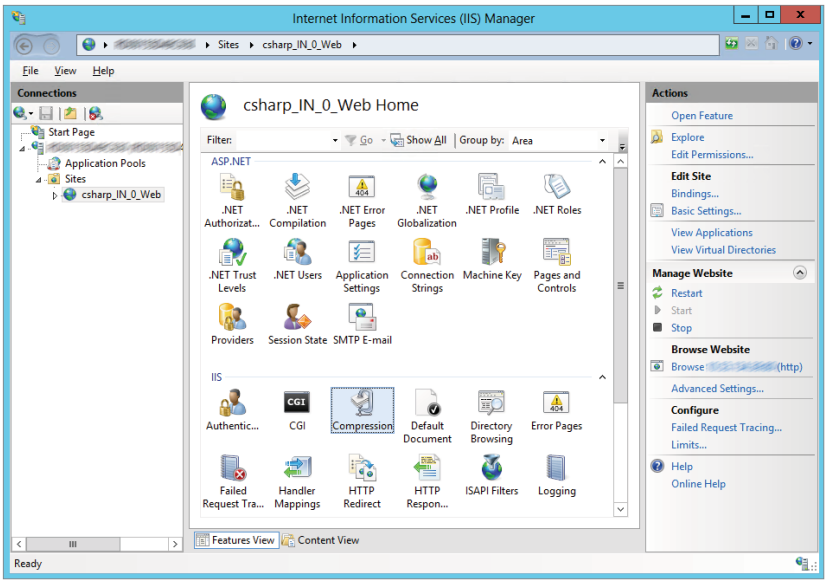


FIGURE 4-22

2. By default, both static and dynamic compression are enabled. As a test, access the ASP.NET MVC 4 Web Role and review the size of each file downloaded from the ASP.NET MVC 4 Web Role hosted on Windows Azure. Figure 4-23 illustrates a Fiddler trace of the request.

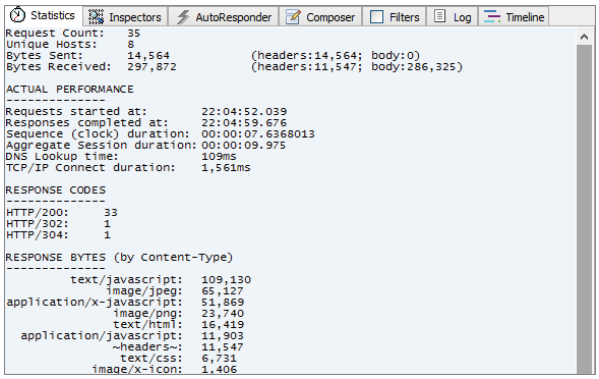


FIGURE 4-23

NOTE *Pay special attention to the* `Bytes Received` *value and the values present in the* `RESPONSE BYTES` *(by* `Content-Type`*) table. These change significantly when compression is modified.*

www.it-ebooks.info

3. Return to your Remote Desktop Connection, and deselect the static and dynamic compression check boxes. Then select the Apply link in the Action pane, as shown in Figure 4-24.

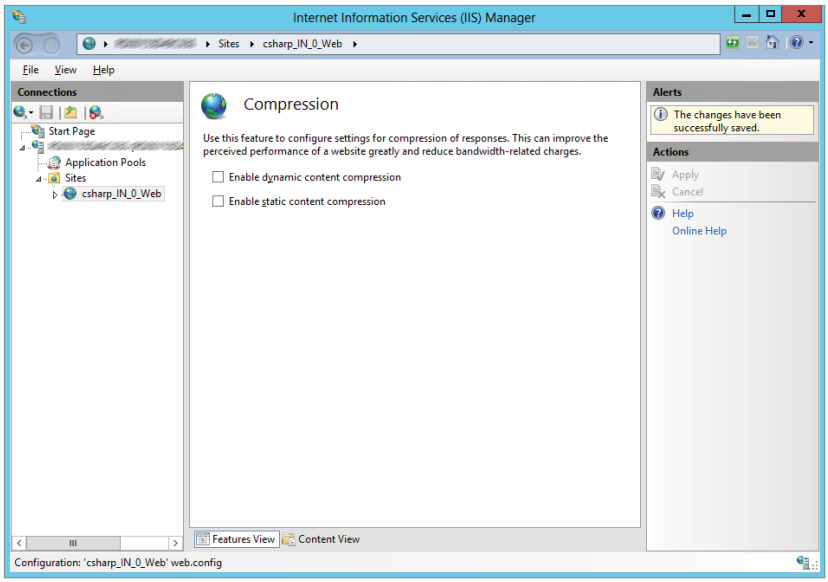


FIGURE 4-24

4. After compression has been disabled, make another request to the ASP.NET MVC 4 Web Role while tracking it in Fiddler and compare it with the values from step 2. You will notice a significant change in size, as illustrated in Figure 4-25.

5. Check the Statistics tabs for both requests, compressed and uncompressed to view the complete time taken to request and render the web page.

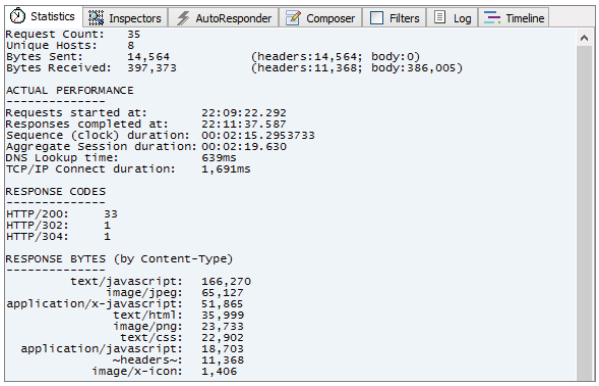


FIGURE 4-25

www.it-ebooks.info

Changing the Output Caching

In this section, you make some changes to the Output Caching capabilities so you can understand what the feature does. For example, output caching can be used to store dynamic content in memory so that the script required to generate each request does not need to be executed for each page. This can yield some significant performance improvements.

NOTE *This section requires you to make a Remote Desktop Connection. For full instructions on how to do this, refer to Chapter 8.*

---

1. In Visual Studio 2012, open the ASP.NET MVC 4 Web Role and then open the View\Shared_Layout.cshtml file.
2. Add the code shown in Listing 4-17 to the bottom of the file, just before the closing `</body>` tag.

LISTING 4-17: Add a DateTime Stamp to the _Layout.cshtml File

```
<div>@DateTime.Now</div>
```

3. Make a Remote Desktop Connection to the ASP.NET MVC 4 Web Role, as discussed in Chapter 8, and open the IIS management console.
4. Navigate to and select the Views\Shared_Layout.cshtml file, as shown in Figure 4-26.

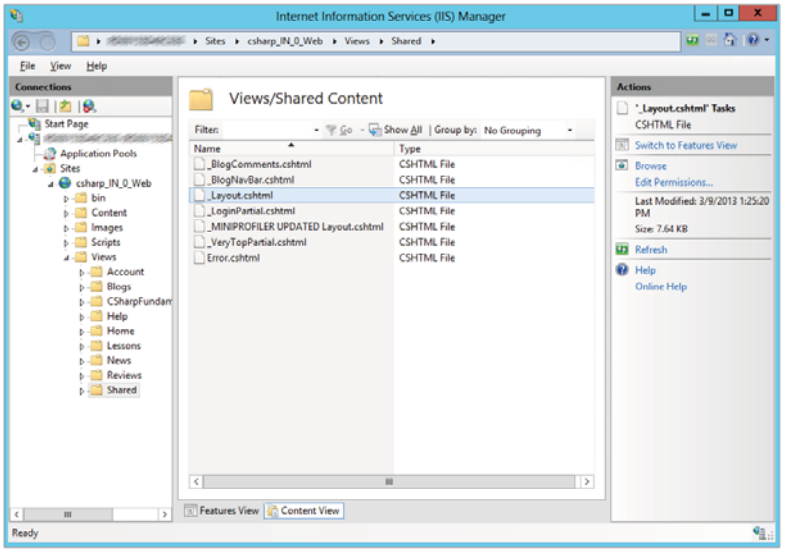


FIGURE 4-26

www.it-ebooks.info

5. When selected, click the Features View tab on the lower-middle section of the IIS management console. Then double-click the Output Caching icon, as shown in Figure 4-27.

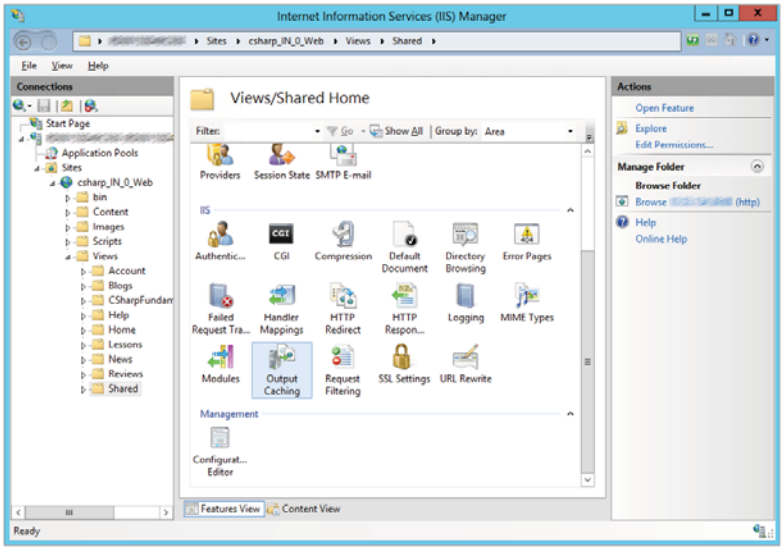


FIGURE 4-27

6. Select the Add link on the Action pane. The Add Cache Rule window opens, as shown in Figure 4-28.

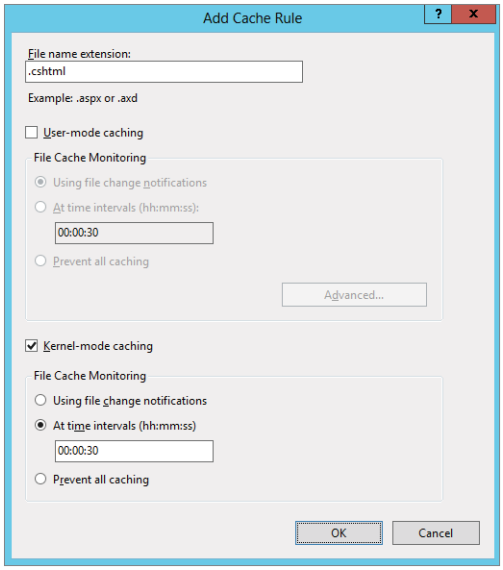


FIGURE 4-28

www.it-ebooks.info

7. Enter the configurations as shown in Figure 4-28, and select OK.
8. Republish the ASP.NET MVC 4 Web Role so that the `DateTime` stamp code modification can be rendered.

NOTE *Chapter 6 is where the details are provided for publishing a Windows Azure Web Role. You might want to complete this chapter before continuing with this example.*

---

9. Access the main page of the website, select F5 multiple times, and note the `DateTime` stamp does not change for 30 seconds, as per the configuration.

The fact that the time stamp does not change for 30 seconds means that the page is being cached and therefore is not downloaded, saving bandwidth and increasing performance. In most cases, you would set the file-cache time interval to a value greater than 30 seconds. The actual value is specific to the application hosted on the web site. It is possible for some files not to change for days or weeks. Therefore they do not need to be downloaded again for a longer period of time. An understanding of an application's purpose is required for setting the proper value for this setting.

COMPARING ASP.NET MVC 4 PERFORMANCE AFTER TUNING

Recall from Table 4-1 that baseline performance metrics were captured. These metrics measured the speed and size of files required to render requests made to the following:

- The **ASP.NET Website** at `http://aspnet.thebestcsharpprogrammerintheworld.com/blogs/Using-the-as-keyword-versus-boxing.aspx`
- The **ASP.NET MVC 4 Web Role** at `http://mvc-4.cloudapp.net/Blogs/BlogLocator/100`

From the captured baseline metrics, the optimizations implemented included:

- Leveraging browser caching
- Enabling compression
- Serve scaled images
- Optimizing images
- CSS and JavaScript minification
- Bundling of CSS and JavaScript files
- IIS compression and caching

When you learn how to deploy your ASP.NET website and ASP.NET MVC 4 Web Role to the Windows Azure platform discussed in Chapter 5 and how to execute it in Chapter 6, you can

www.it-ebooks.info

redeploy the code enhancements made in this chapter to test and review the improvements. Even better, after testing, you can go straight into the production environment with the most optimal code.

After you deploy the optimized code, you can capture the statics in the same way you captured them when you created the baseline in this chapter. Table 4-6 and Table 4-7 provide the performance metrics after the recommendations presented in this chapter were implemented. Notice that, in almost every case, there was an improvement in speed and, in many cases, a decrease in size.

TABLE 4-6: ASP.NET Website Optimized Performance

ATTRIBUTE	VALUE
# of HTTP requests (objects)	25–58
Total size (bytes)	194 K–623.6 KB
Total download time (seconds)	2.15–4.39
HTML size (bytes)	33.4-62.5
HTML download time (seconds)	0.25
CSS size (bytes)	8.12
CSS download (seconds)	0.24
Image size (bytes)	56.3–89.3
Image download (seconds)	3.4
Script size (bytes)	42.1–385.8
Script download (seconds)	0.97
Total header sizes (bytes)	23.6 KB

TABLE 4-7: ASP.NET MVC 4 Web Role Optimized Performance

ATTRIBUTE	VALUE
# of HTTP requests (objects)	11–28
Total size (bytes)	134 KB–332 KB
Total download time (seconds)	2.7–3.1
HTML size (bytes)	4.6–36.3
HTML download time (seconds)	1.44

www.it-ebooks.info

CSS size (bytes)	1.62 KB
CSS download (seconds)	0.19
Image size (bytes)	26.1 KB–87 KB
Image download (seconds)	1.41
Script size (bytes)	66.4–242.9 KB
Script download (seconds)	1.37
Total header sizes (bytes)	14.5

SUMMARY

In this chapter, you captured a set of baseline performance metrics using tools such as Fiddler, MiniProfiler, and the F12 Developer Tool Suite in Internet Explorer. It's important that you capture the baseline so that you have something to which you can compare your post-optimization performance.

You also used Google PageSpeed to help identify some inefficiencies in the sample ASP.NET website, some of which were referencing an image. You then modified the `width` and `height` attributes to reduce the size instead of just reducing the image size. You also ensured that browser caching was implemented effectively.

In addition, you learned that by minifying and bundling CSS and JavaScript files, you could realize gains in performance in both ASP.NET websites and ASP.NET MVC 4 Web Roles.

After you implemented these optimization examples and redeployed the enhanced source code to the respective Windows Azure environments, you saw positive results. In almost every case, there was an improvement in download speed and a reduction in overall download size.

www.it-ebooks.info

PART III
Deployment

- **CHAPTER 5:** Discussing ASP.NET MVC 4 Windows Azure Deployment Techniques
- **CHAPTER 6:** Deploying an ASP.NET MVC 4 Project to Windows Azure

www.it-ebooks.info

5 Discussing ASP.NET MVC 4 Windows Azure Deployment Techniques

CONCEPTS

IN THIS CHAPTER

- Preparing your application for the Windows Azure Platform
- Examining Developer Centers and supported SDKs
- Introducing Cloud Computing Services
- Accessing the Windows Azure Platform
- Choosing your Windows Azure Services
- Discussing deployment options
- Planning database migration and storage
- Monitoring the status of a deployment

To this point, the focus of this book has been on migrating an ASP.NET website to an ASP.NET MVC 4 application and optimizing it. New topics learned during this reinvention included models, views, controllers, RouteMaps, the Razor scripting language, MiniProfiler, and the F12 Developer tools available in Internet Explorer. These concepts and tools advanced the example blog website for this book from older to newer technology and models. After utilizing some online tools such as Modern.IE or Google PageSpeed and incorporating those tools' suggestions, the ASP.NET MVC 4 application can now perform and respond much faster. With that, the ASP.NET MVC 4 application for this book is complete and ready for you to host so that people can find, use, and share the information on the blog.

Besides the cloud, you can use a variety of platforms, hosting types, and services to deploy the ASP.NET MVC 4 application. For example, you can create a co-hosted website with a web hosting provider, build your own server, and then connect it to the Internet, or rent a server or virtual machine from an outsourcer.

www.it-ebooks.info

In this chapter you will learn how to prepare an application for deployment and hosting on the Windows Azure platform. This chapter discusses topics such as understanding the supported SDKs like PHP, Ruby, Java and so on, as well as understanding the different types of Cloud Services, such as IaaS, PaaS and SaaS (defined later in the chapter). Additionally, the chapter covers the different deployment options and how to monitor the status of a deployment.

Because you have a number of hosting options, you should exercise due diligence in researching what works for your situation before making the final decision on where to host your website. The next section provides some tips for determining if Windows Azure is the best platform for your system as well as the sample ASP.NET MVC 4 application for this book.

NOTE *This chapter contains key deployment concepts. For a step-by-step guide on how to implement these concepts, see Chapter 6.*

PREPARING YOUR APPLICATION FOR THE WINDOWS AZURE PLATFORM

To evaluate whether the Windows Azure platform suitably meets your needs (and the needs of the ASP.NET MVC 4 sample application), you should consider the benefits it offers, which include but are not limited to those discussed in the following sections.

Straightforward Implementation

Without much effort, a company can get its system implemented and running in a live environment on the Windows Azure platform faster than ever before — without purchasing hardware or software licenses, or implementing services, such as a Release Manager. You simply gain access to the Windows Azure environment, create the website or Web Role and/or storage feature, and, with a little configuration, the system is ready for use.

Scalability, Availability, and Durability

As mentioned in more detail in Chapter 3, a benefit of the Windows Azure platform is that it scales quickly and easily. The Windows Azure platform has something called *elastic scalability*, which allows you to quickly add more physical resources when your system requires them, such as an increase in traffic. Then, when the burst of traffic decreases, you can remove the additional capacity.

Releasing Internal Resources

In addition, by having multiple instances of a Windows Azure Web Role, if one were to fail due to a worker process hang, for example, you can deploy a new role and automatically direct

www.it-ebooks.info

traffic to the working instance. From a data storage perspective, Windows Azure replicates storage services to multiple different servers to guarantee availability and durability.

Quality Support/Experienced Practitioners

Another benefit is that you can release internal IT resources to focus on tasks and activities that are most critical to business and innovation. Because management of infrastructure is increasingly outsourced, the in-house IT department no longer attends to network, OS patching, system issues, and other duties. These outsourced responsibilities are thus managed by high-quality support engineers and top experienced practitioners who proactively work to quickly respond to the technical issues happening on the platform.

NOTE *The Microsoft Assessment and Planning Toolkit can help you plan your migration to Windows Azure. You can search for it using Bing or access it directly at this URL:* `http://technet.microsoft.com/de-de/solutionaccelerators/dd537566.aspx`

Mobility

Azure's mobility enables you to access files, notes, intellectual property, and e-mail from a large variety of devices from almost any location in the world. For example, Microsoft's SkyDrive or Office 365 are cloud applications hosted on the Windows Azure platform — both of which enable a company or an individual to store, share, and collaborate on tasks from anywhere on any device. You get all of this is at a price that is likely to be lower than the cost of hosting your own server or managing and running your own data center.

Reduced Cost

Cost is a significant factor when deciding which hosting service or model to use for your website. If your website does not get significant traffic or generate revenue, most likely the Windows Azure platform or any of the cloud platforms are not the most affordable or logical option to host your website. For these instances, some web-hosting companies have only a few dollars ($US) a month for hosting. Therefore, if you are an individual, the cheapest solution is likely the best option.

UNDERSTANDING DEVELOPER CENTERS AND SUPPORTED SDKS

Because ASP.NET MVC 4 is based on the .NET Framework, it is certain to be supported when you deploy to the Windows Azure platform. At minimum, you must install the .NET Framework 4.0 for an ASP.NET MVC 4 application to function as expected. As shown on the Windows Azure Web Site and illustrated in Figure 5-1, you can download and use a number of

www.it-ebooks.info

other SDKs to code your application. By using the frameworks provided in the SDKs, you can be confident of seamless integration and support when you deploy to a Windows Azure Web Site, Web Role, or virtual machine.

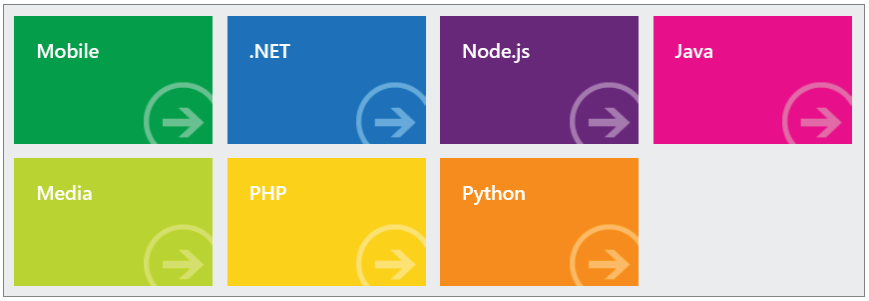


FIGURE 5-1

Each of the Developer Centers provides information for the selected technology. As a result, .NET, Node.js, Java, PHP, and Python are currently supported on the Windows Azure platform. Additionally, Mobile Services and Media Services have tutorials and SDKs to implement those features available on the website. You can also find tutorials, reference documents, API, and REST interface descriptions for the website's technology framework.

NOTE *You can access the Getting Started and Developer Center SDKs at this URL:* `https://www.windowsazure.com/en-us/documentation`.

It is important to confirm that the technology and components your website requires to function and operate properly on the Windows Azure platform. By using these SDKs in preparation for deployment and performing a proof of concept early during your project, you can avoid surprises and costly delays. *Proof of concept* is a small program that you use to test the basic features of a software library before you make a large scale commitment to it.

INTRODUCING CLOUD COMPUTING SERVICES

Before moving into the cloud, you need to make two specific decisions. The first is determining the service model on which you'll host your application, for example IaaS, PaaS or SaaS. Each of these service models requires different levels of support and responsibilities once a system is deployed to the platform. These differences are discussed in detail in later sections of this chapter. The second decision is determining in which deployment model you will host those service models. Do you want a private or on-premises cloud, or do you want to deploy your application directly into a public cloud? Security and cost are relevant things to consider for these deployment models, which are discussed in later sections of this chapter.

www.it-ebooks.info

Understanding Cloud Computing Service Models

Different Cloud Services have different levels of technological support. These Cloud Services offer the resources required to operate most web-enabled systems imaginable. Cloud Services give you the hardware and the operating systems to run the website or solution as well as the binaries for executing and compiling your code. Other features are the Network connections and bandwidth that funnel information to your customer. Table 5-1 illustrates some of the most common Cloud Computing Service models.

TABLE 5-1: Common Cloud Computing Service Models

ACRONYM	DESCRIPTION	SHORT DEFINITION
IaaS	Infrastructure as a Service	Provides physical or virtual machines for service hosting.
PaaS	Platform as a Service	Computing platform with OS, programming language execution, database, and websites.
SaaS	Software as a Service	Cloud providers install software that users can access and use from any location, for example, Microsoft Office 365.
DaaS	Database as a Service	A database for systems to connect to and use.
NaaS	Network as a Service	VPN, bandwidth on demand, and network/transport connection services.

The most common Cloud Computing Services, all of which are supported in the Windows Azure environment are IaaS, PaaS, and SaaS. As illustrated in Figure 5-2, each of those services are built on top of each other and sometimes referred to as the Cloud Stack.

NaaS, recently released on the Windows Azure platform, allows you to now configure Virtual Private Networks, site-to-site networks and point-to-site networks. DaaS is deployed and used by creating a database in the cloud and connecting to it. All the previously mentioned service models are available on the Windows Azure platform.

What is IaaS?

At the bottom level of Figure 5-2, you'll notice *Infrastructure as a Service* (*IaaS*) which is the most basic model level. IaaS provides the physical hardware where cloud users can install operating systems and/or applications that run on this hardware. In most cases, it contains the virtual machines managed by products, such as Hyper-V or VMware. As previously discussed, virtual machines support the elastic scalability needed for system usage bursts. Other resources commonly provided with IaaS may include IP addresses, firewalls, and file storage.

www.it-ebooks.info

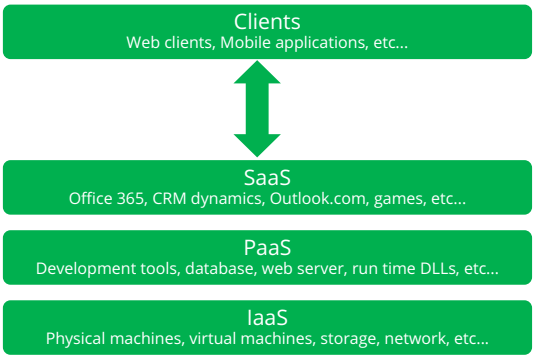

Clients
Web clients, Mobile applications, etc...
SaaS
Office 365, CRM dynamics, Outlook.com, games, etc...
PaaS
Development tools, database, web server, run time DLLs, etc...
IaaS
Physical machines, virtual machines, storage, network, etc...

FIGURE 5-2

What is PaaS?

The operating system exists on top of the physical hardware or virtual machine; this is where the *Platform as a Service* (*PaaS*) resides. Besides operating systems (such as Windows Server 2008 R2, Windows Server 2012, or Linux), code execution binaries, web servers, and databases provide environments where individuals, developers, or companies can host their computer system. The PaaS is also where the ASP.NET MVC 4 sample website (used throughout this book) exists. Both the Windows Azure Web Site and Windows Azure Cloud Service provide the PaaS services.

NOTE *For more on how the ASP.NET MVC 4 sample operates on the PaaS, see Chapter 6.*

What is SaaS?

Cloud providers offer *Software as a Service* (*SaaS*) so consumers can access and use software on numerous devices from any location provided they have an Internet connection. The software is typically installed, supported, and managed on the PaaS environment by the cloud provider thereby eliminating those responsibilities from the individual or company. Some examples of SaaS are Microsoft Office 365, Microsoft SkyDrive, and Microsoft CRM Dynamics Online.

Understanding Deployment Models

In addition to the Cloud Computing Service Models there are also different types of deployment models, which are summarized in Table 5-2. These are discussed in more detail in the following sections.

www.it-ebooks.info

TABLE 5-2: Cloud Deployment Models

DEPLOYMENT TYPE	DESCRIPTION
Private cloud	Cloud infrastructure for a single organization
Public cloud	Cloud infrastructure made available by a provider, that is, Windows Azure
Community cloud	Cloud infrastructure shared between two or more organizations
Hybrid cloud	An arrangement of two or more of the other deployment models

Understanding Private and Public Clouds

The primary difference between a private and public cloud is as follows:

- **A private cloud** infrastructure is owned, operated and managed by the same company who owns the systems hosted in that cloud. You have the further option of having either an offline versus an online private cloud.
- **A public cloud** is typically owned and managed by a cloud hosting company that allows external businesses or individuals to host and manage their applications on their platform for a fee. Freeing the business and individuals of the management of the cloud infrastructure. The following sections discuss each.

The following sections examine private and public clouds in more detail.

What Is a Private Cloud?

A *private cloud* is likely utilized and implemented by companies or governments that must perform actions required by law or for regulatory compliance. These capabilities may not yet be possible in the public cloud. For example, a private cloud in the Windows Azure environment located onsite, resides in a company's or organizations own private data center. The benefits of a private cloud are that you can customize it however you want, and it is perceived to be more secure compared to a public cloud because you have more control over the environment. Contrast that with a public cloud where you do not control operating system patches, the hardware, or the infrastructure, such as load balancers, network, and firewalls.

Offsite versus Onsite Private Clouds

You can have either an offsite or onsite private cloud. The main difference between the two is that the offsite infrastructure is hosted in an outsourced data center (see Figure 5-3):

- With an *offsite private cloud*, the company owns and supports all the infrastructure required to run the cloud. Because the offsite private cloud is a cloud environment in an outsourced data center, it is not shared with other companies or customers. A business might decide to host their system in an offsite cloud instead of building

www.it-ebooks.info

and managing their own data center facilities. This results in a lower cost of IT ownership and maintenance because the data center owner provides the building, power supplies, and network infrastructure.

- An *onsite private cloud* is maintained by the company that owns the facilities in which it operates.

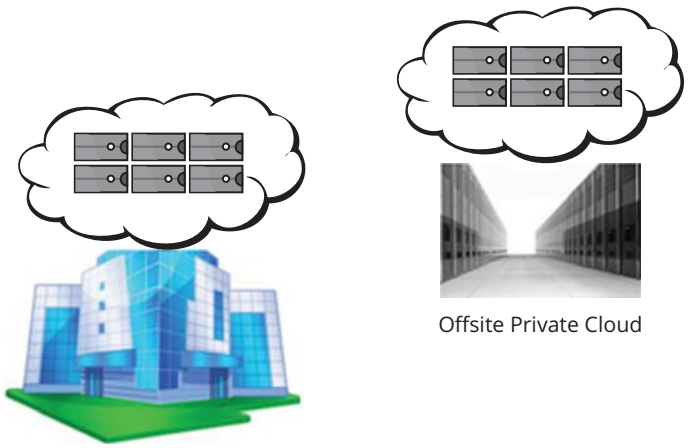


FIGURE 5-3

What is a Public Cloud?

A *public cloud* is a large group of servers owned, operated, supported, and maintained by a cloud provider. When individuals or companies sign up for a Windows Azure account and deploy a website or Web Role, it is deployed into a public cloud.

Pros and Cons of a Private versus a Public Cloud

Figure 5-4 illustrates a comparison or pros and cons between a private and public cloud. Full control of the environment, the usage of a private network, and having full control of and access to the content (that is, more security) are positives of the private cloud. Whereas initial capital costs, the requirement of building and managing the environment, and not getting many synergies by sharing the capacity are the private cloud deployment model's downsides.

Don't get the wrong impression that placing content or intellectual property into a public cloud is not secure. The private cloud does give complete control of access to the company owning and operating the private deployment model; however, even the public cloud has sophisticated layers of security.

www.it-ebooks.info

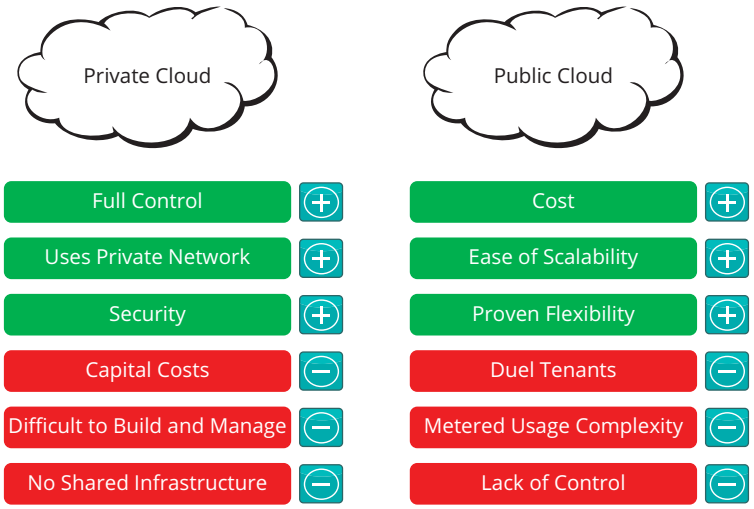


FIGURE 5-4

The positive aspects of the public cloud deployment model are that it is less expensive to deploy and run; scaling the environment does not require the purchase and configuration of any hardware; and there are many real-world examples of high-performing, high-available solutions deployed to the public cloud that you can use as models. Negative aspects include the possibility you may be sharing your environment with other customers or companies and that this somehow will impact your websites' performance or availability. Also, initially, the cost of the deployment may be complicated to calculate, and you do not have complete control over operating system patches or if other third-party components get installed. For all these reasons, you should use the private cloud only if you have a legitimate business justification.

Understanding a Community Cloud

A *community cloud* is a group of servers shared by a group of companies or individuals.

This approach is usually performed with partners or individuals who want to gain more control over their environment while spreading the costs across a group of trusted partners in the community. An advantage to this model is that you reduce the costs incurred by the purchase of hardware and maintenance. A disadvantage is that you share the environments with other systems that may impact the performance of your system by either using too much of the systems resources or by installing or updating dependencies that have a negative impact on the co-hosted systems.

www.it-ebooks.info

Understanding a Hybrid Cloud

A hybrid cloud is generally only used by enterprises who want to build a solution that is not yet completely available in the cloud. Figure 5-5 shows a hybrid deployment model that implements two or more of the four described deployment models. This model means you can control areas that you do not want outsourced but rather want to keep in-house or on-premises, while at the same time placing the pieces you want to make public into a multi-tenant cloud. You can also realize some cost advantages by placing content you still consider intellectual property on the public cloud.

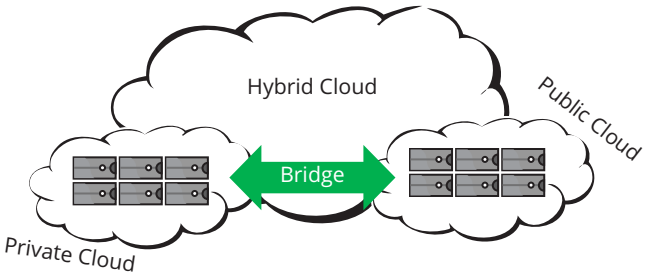


FIGURE 5-5

A hybrid cloud is more cost effective when compared to a private cloud because your internal IT department only manages, supports, and scales out a portion of the company's solution. You can pick and choose the best components of each deployment model, and build your own cloud solution. The down side is that because the system is hybrid, it's also unique and you'll have no best practice examples or patterns to follow. If you choose a hybrid cloud deployment model, you innovate and build a solution, which may result in unique problems that might need skilled and experienced cloud experts to work through and support. Therefore, you must choose your solution carefully and consider supportability and flexibility options before committing.

ACCESSING THE WINDOWS AZURE PLATFORM

At this point the different Cloud Service and deployment models, plus the supported SDKs should be understood. The next step is to gain access to the Windows Azure platform. This, of course, assumes that you have decided to deploy to a public cloud and chose either an IaaS, PaaS, or SaaS Cloud Service. There are currently no private, hybrid, or community deployment models on the Windows Azure platform.

Obtaining access to the Windows Azure platform is not complicated. The details of this are discussed and performed in Chapter 6. After you provide the required information to access the platform, you log into the Windows Azure Portal, and can view the main overview page similar to that shown in Figure 5-6.

www.it-ebooks.info

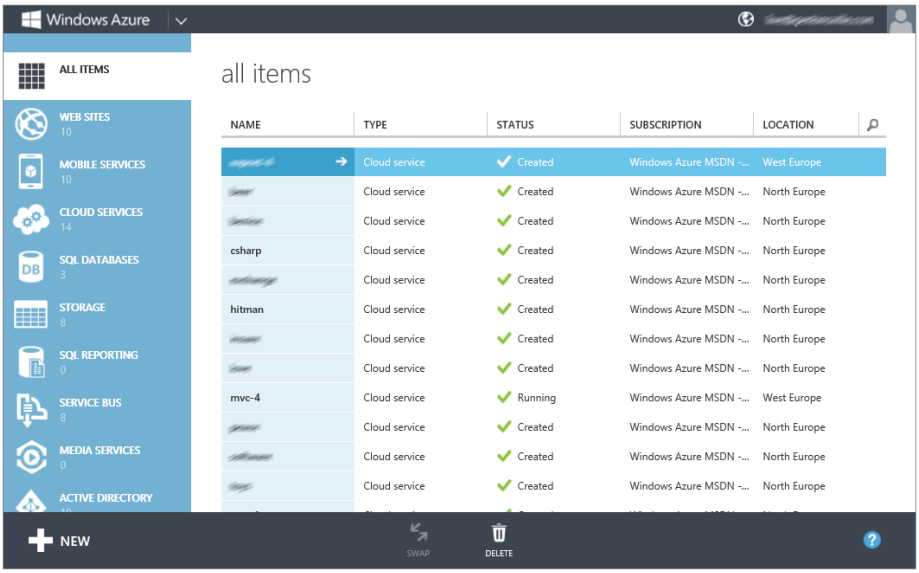


FIGURE 5-6

The right side of the management pane contains various platform items. You can click a group to access a specific item, or click the +NEW link to create a new item of any type. Table 5-3 shows the various services you have available to you.

TABLE 5-3: Windows Azure Item Services

COMPUTE	DATA SERVICE	APP SERVICE	NETWORK
Web Sites	SQL Database	Service Bus	Virtual Networks
Virtual Machines	Table Storage	Media Services	
Mobile Services	HDInsight (new)	Active Directory	
Cloud Services	SQL Reporting		

Most of the services are intuitive and do as they are called, such as Web Sites, SQL Database, and Cloud Services, all of which have examples and exercises provided in Chapter 6. For many, Service Bus, Active Directory, SQL Reporting, and others likely need some additional investigation and study. All the details and descriptions for these services are provided at the following URL; click the Services link as shown in Figure 5-7: `https://www.windowsazure.com/en-us/documentation`. There is also a discussion of some Windows Azure Services provided in the following section, "Choosing your Windows Azure Services." Nonetheless, a brief explanation of each Windows Azure service item is provided in the following list.

www.it-ebooks.info

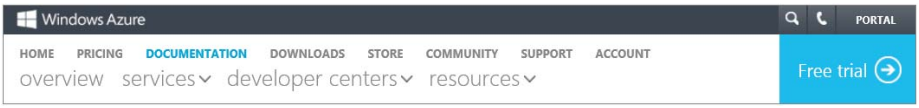


FIGURE 5-7

- **Compute:**
  - **Web Sites:** An example for creating a Windows Azure Web Site is provided in Chapter 6. When you create a website in Windows Azure, it is co-hosted with other websites on a single virtual machine.
  - **Virtual Machines:** An example of Window Azures IaaS feature. It is similar to creating a virtual machine using Hyper-V or VMware, however it is now done in the cloud.
  - **Mobile Services:** Provides a back-end for your mobile app. If you have created, for example a Windows Phone app, and need to store some data, this service is quick and easy to setup and configure for just that purpose.
  - **Cloud Services:** An example of Window Azure's PaaS feature. You do not have control or responsibility over the infrastructure, but can choose from multiple operating systems like Windows Server 2008R2, Windows Server 2012 or many versions of Linux. An exercise of creating a Cloud Service is provided in Chapter 6.
- **Data Service:**
  - **SQL Database:** A Windows Azure DaaS feature, this is a full SQL Server database instance hosted in the cloud. A detailed description of how to create and utilize a SQL Server database in the cloud is provided in Chapter 6.
  - **Table Storage:** A storage structure that does not require complicated data structures but does required very large storage resources.
  - **HDInsight:** Windows Azure Big Data service feature. This is a new industry with many unknown opportunities. You can use this feature to learn more about Big Data analysis with tools like Hadoop.
  - **SQL Reporting:** SQL Server reporting services is a platform for quick creation and execution of reporting commonly used for sales and marketing forecasting.
  - **Recovery Services:** Tools for backing up and recovering your application hosted on the Windows Azure platform.
- **App Service:**
  - **Service Bus:** Is similar to MSMQ or Tuxedo in that is manages the flow of requests from clients to a backend system.

www.it-ebooks.info

- **Media Services:** Provides a platform for multicasting and video streaming.
- **Active Directory:** A location for creating domains and authenticating users in the cloud, similar to an Active Directory (AD) hosted in a standard enterprise. Connection between the enterprise's AD with an AD in the cloud is supported.

- **Network:**
  - **Virtual Networks:** You can create a standalone network in the cloud with virtual machine and manage it just as any corporate or personal network. Virtual Network supports site-to-site, point-to-site, and VPN tunneling.

This section did not cover the details of all the services and options available on the Windows Azure platform because the best way to understand your options is to access the platform and start testing them, working with them, and obtaining practical experience. Learning how to apply the services to your company and communicating those benefits internally can come only after using and working in the platform over time. Some very good documentation that discusses all the Windows Azure service can be found here: `www.windowsazure.com/en-us/documentation/`.

CHOOSING YOUR WINDOWS AZURE SERVICES

For the examples and exercises provided in this book, the Compute and Data Services are chosen for you. However, when you're choosing which services you require for a system other than the example, you may need more information. For example, what are the differences between a Windows Azure Web Site, Cloud Service, and Virtual Machines, and when should you use which? What is the difference between a Windows Azure SQL Database and Table Storage? And what are SQL Reporting, HDInsight, and Active Directory?

Using Azure Web Site versus Cloud versus Virtual Machine

Let's begin with the analysis and comparison of the Windows Azure Web Site, Cloud Service, and virtual machine. With a Windows Azure Virtual Machine, you can do basically anything you could do with a VMware or Hyper-V virtual machine. For example, you can build the infrastructure completely on your terms, choose the operating system, such as Windows Server or Linux, and securely connect your on-premises system with virtual machines running on the Windows Azure platform.

A Cloud Service, also referred to as a Web Role, provides an easy way to build and extend enterprise-level applications that require high availability and multitier settings. Windows Azure customers can also use Web Roles to create a Software as a Service (SaaS) product for their customers, which customers can use and access from any place in the world. Websites are co-hosted on multitenant instances, meaning they are co-hosted with other websites on the same virtual machine, but neither Remote Desktop Connection nor the ability to run programs

www.it-ebooks.info

nor software in an elevated mode (such as Run as Administrator) are supported. However, websites support FTP and GIT deployment options, which provide for near-continuous deployment strategies. Figure 5-8 illustrates these capabilities in more detail. Note that when running a virtual machine on the Windows Azure platform, you can utilize features available to both a website and a Web Role. Therefore, all the capabilities shown in Figure 5-8 apply to a virtual machine, too.

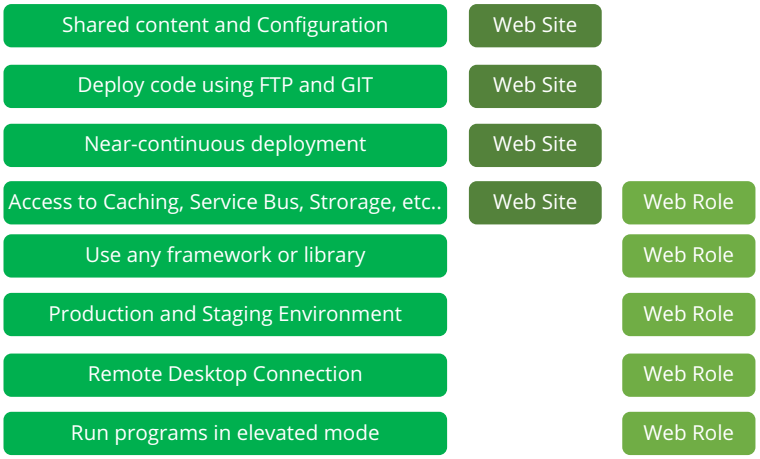


FIGURE 5-8

Understanding Data Storage Features

There are also two data storage capabilities to choose from:

- **The Windows Azure SQL Server database:** This is a fully functional relational database service. You should chose the Windows Azure SQL Server database instance if you require a relational database, stored procedures, TSQL, and want to run ACID Transactions. Many complex enterprise and even some smaller systems have a relational database powering it. However, some simpler systems, requiring only a storage solution that uses key/value pairs, should consider Windows Azure Table Storage.
- **Windows Table Storage:** Using Table Storage, you are not locking down your system to a data model; the data structure is not relational and therefore does not support SQL queries using `JOIN`s or stored procedures, but transactions are supported.

You choose Windows Azure SQL Database when:

- Your database must support rich data types, access to data using joins, or database aggregates such as `MAX`, `MIN`, `AVG`, and so on.
- Your database needs to support stored procedures.

www.it-ebooks.info

- Your database likely will not exceed 150 GB. This is not a limit; it just means that additional actions need to be taken if exceeded.
- Your database is relational with primary and foreign keys linking table together.
- Your application already uses the Microsoft SQL Server database engine.

Choose Windows Azure Table Storage if:

- You need large data storage (multiple terabytes).
- Your database does not have complicated data schemes.
- Your database does not use joins and complicated server-side queries and does not require secondary indexes.

What is Windows Azure SQL Reporting?

Windows Azure SQL Reporting is built on the SQL Server Reporting services. This capability provides access to reports using only an Internet browser. After the report is accessed and rendered, it can be extracted to Excel and saved as a PDF. You can find more information by searching for **Windows Azure SQL Reporting** from any major search engine.

What is HDInsight?

HDInsight is one Microsoft's Big Data initiatives and solutions. Big Data analysis and technologies are all over the news and discussion boards, and it is commonly referred to as a solution without a problem to solve. Nonetheless, most of the major players in the IT industry, such as Amazon.com, Google, IBM, Facebook, Microsoft, and so on are investing big into this technology. If you are interested in working with it, you can add the feature to your Windows Azure platform and give it a test drive.

NOTE *There are a number of articles and labs available for learning HDInsight. You can find more information about Big Data analysis using HDInsight here:* `www.windowsazure.com/en-us/manage/services/hdinsight/`.

Using Active Directory

The Active Directory capabilities available on Windows Azure are similar to those available in a standard domain. The Active Directory supports the creation of domains, user, computer and service accounts, and the authentication of those accounts. This is a feature that is not currently available with any other cloud provider. A significant benefit and usage for this feature is to create and support a single-sign-on (SSO) solution for all your applications that exist in your domain, similar to the Windows Authentication, which uses either NTLM or Kerberos. In many corporations, employees need to know multiple user IDs and passwords to access the

www.it-ebooks.info

many systems in the company to do their job. With an SSO solution, the employee can sign-on once and access all the systems that they have authorization to as long as these resources exist in the same domain. This is one of the more exciting features available in the Windows Azure platform, and is full of innovative possibilities.

Now that you know how to prepare your application for Windows Azure, which SDKs are supported, the different types of Cloud Services (PaaS, IaaS, and SaaS), and the differences between a Web Site, Web Role/Cloud Service, and a virtual machine, it's now time to review the different deployment methods. The next section covers the many options in detail.

UNDERSTANDING DEPLOYMENT OPTIONS

The Chapter 6 exercises and examples have steps that walk you through using FTP, GIT, and TFS to deploy the sample ASP.NET MVC 4 project to both an Azure Web Site and a Web Role. There are a few other deployment options, and new ones are added often. A current list of available deployment options include:

- Team Foundation Service
- CodePlex
- GitHub
- Dropbox
- Bitbucket
- Local Git

Integrating Source Control with a Cloud Service

Having a source code repository is important for a number of reasons. Primarily, the repository stores and secures source code, which is likely one of the most valuable sources of intellectual property that a company has. Additionally, many source code repositories have version controls that allow you to roll back from one version to another as well as simple comparison tools to see how code changes between different versions. Source code repositories are, thus, a recommended and commonly followed practice. Finally, the source code repository is the location where Release Managers get the final version of a release that's eventually moved into the production environment for the customers and visitors to use.

The basic steps for integrating the source control with a Cloud Service are as follows:

1. Select the Cloud Service with Windows Azure management console.
2. Under the Integrate Source Control header, click the Set Up TFS Publishing link, shown in Figure 5-9.

www.it-ebooks.info

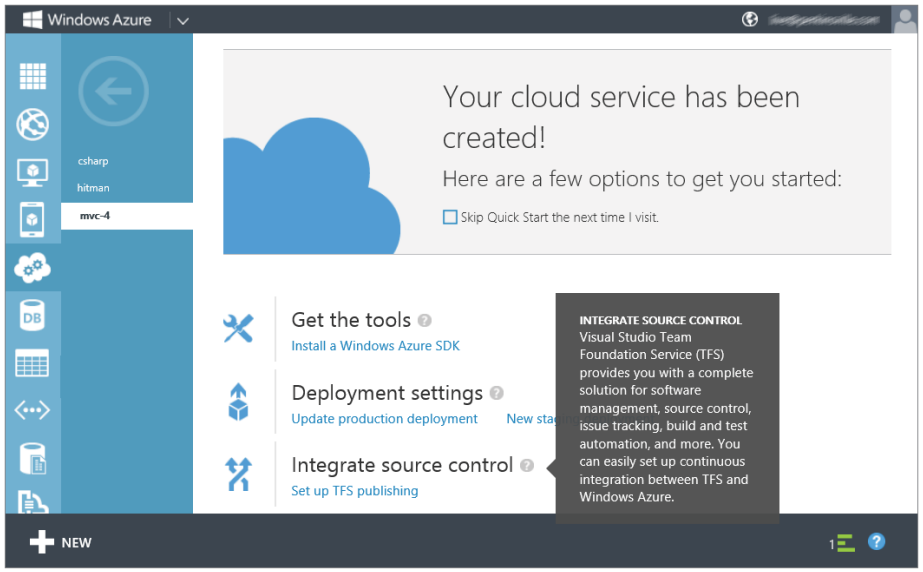

Windows Azure
csharp
hitman
mvc-4
Your cloud service has been created!
Here are a few options to get you started:
Skip Quick Start the next time I visit.
Get the tools
Install a Windows Azure SDK
Deployment settings
Update production deployment
Integrate source control
Set up TFS publishing
INTEGRATE SOURCE CONTROL
Visual Studio Team Foundation Service (TFS) provides you with a complete solution for software management, source control, issue tracking, build and test automation, and more. You can easily set up continuous integration between TFS and Windows Azure.
NEW

FIGURE 5-9

3. Follow the wizard and link an existing TFS account to this Web Role or create a new account.

NOTE *A more complete example of how to integrate a source control with a Cloud Service is provided in the following chapter. In this example, after you create the link, you can deploy and compile the migrated source code, and make it available to your users from the online TFS website capabilities or from Visual Studio 2012.*

Integrating Source Control with an Azure Web Site

To integrate source control with an Azure Web Site, you must select a preferred source control repository. You have several options for doing this including Team Foundation Service, CodePlex, and GitHub.

Although this section only covers the CodePlex and Dropbox source code repositories, the principles illustrated in the following sections are the same for the remaining deployment options.

NOTE *Some options do not support or provide source code management capabilities that TFS, Git, and CodePlex have. Many of the options are simply a portal or staging area from which to deploy code to the Windows Azure Web Site, similar to FTP. Exercises for integrating GitHub and TFS with Windows Azure are provided in Chapter 6.*

---

www.it-ebooks.info

Using CodePlex

You can use CodePlex to create new projects to share with the world or join others who have already started their own projects. Basic steps follow for integrating your source control with a Windows Azure Web Site utilizing CodePlex.

NOTE *The Windows Azure Platform is described as "Microsoft's open source project hosting website. You can use CodePlex to create new projects to share with the world or join others who have already started their own projects."*

---

1. Access the website from the Windows Azure management portal.
2. Click the Set Up Deployment from Source Control link under the Integrate Source Control heading, and the window shown in Figure 5-10 appears.

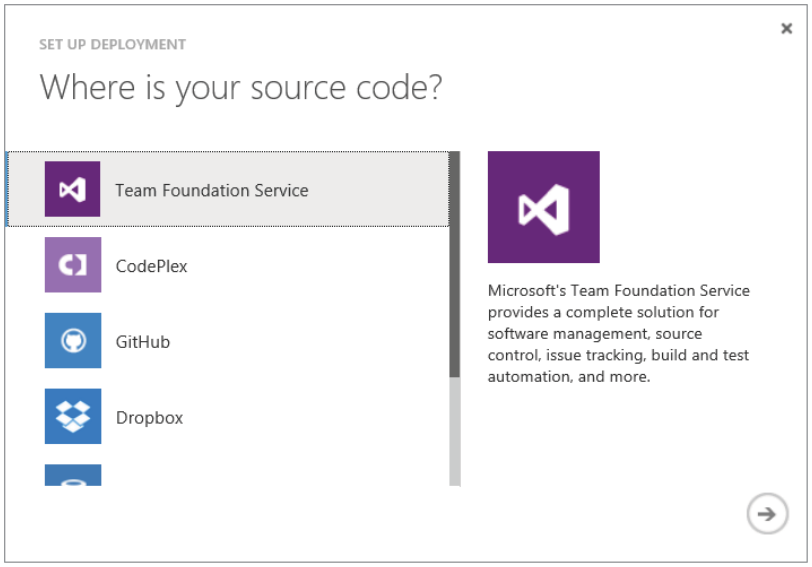

SET UP DEPLOYMENT
Where is your source code?
Team Foundation Service
CodePlex
GitHub
Dropbox
Microsoft's Team Foundation Service provides a complete solution for software management, source control, issue tracking, build and test automation, and more.

FIGURE 5-10

3, Select the preferred source control repository and follow the wizard to configure it. For this example CodePlex is used. When you select CodePlex and then click the Next Arrow, a window opens (Figure 5-11) that requests your CodePlex credentials or allows you to register for the source code repository hosting service. After the link is created, you can deploy your source from CodePlex.

www.it-ebooks.info

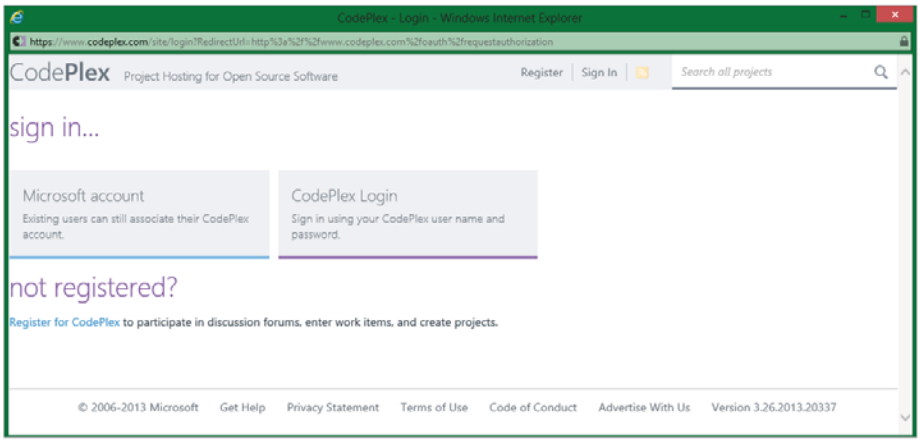


FIGURE 5-11

Using Dropbox

Like Microsoft's SkyDrive, after the Dropbox software is installed on your PC, it acts like a mapped drive or folder accessible using Windows Explorer, as illustrated with Figure 5-12. Windows Azure describes this software as follows:

> *"Dropbox is a free service that lets you bring all your photos, docs, and videos anywhere, and share them easily. Use Windows Azure to quickly sync and deploy your code from your Dropbox folder."*

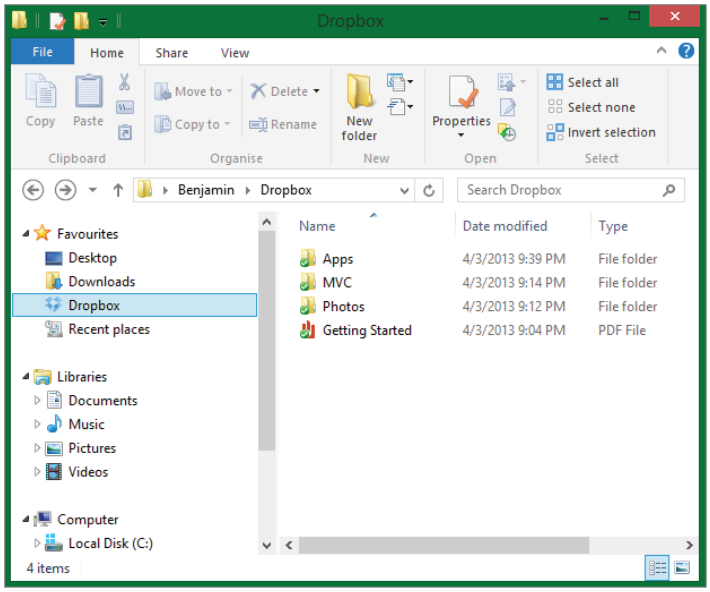


FIGURE 5-12

www.it-ebooks.info

To configure Dropbox as your source code repository, follow these steps:

1. Select the Dropbox from the pop-up, as shown previously in Figure 5-10, and select the Next arrow. The Connection Wizard navigates you through the process of binding the Dropbox account with your Windows Azure account.
2. When you're finished with the Wizard, click the Allow button, as shown in Figure 5-13. When authenticated, a folder called Apps is created in your Dropbox account.
3. Place your code into this directory, and select the Sync button to deploy the code to the Windows Azure Web Site if for some reason it doesn't kick off automatically.

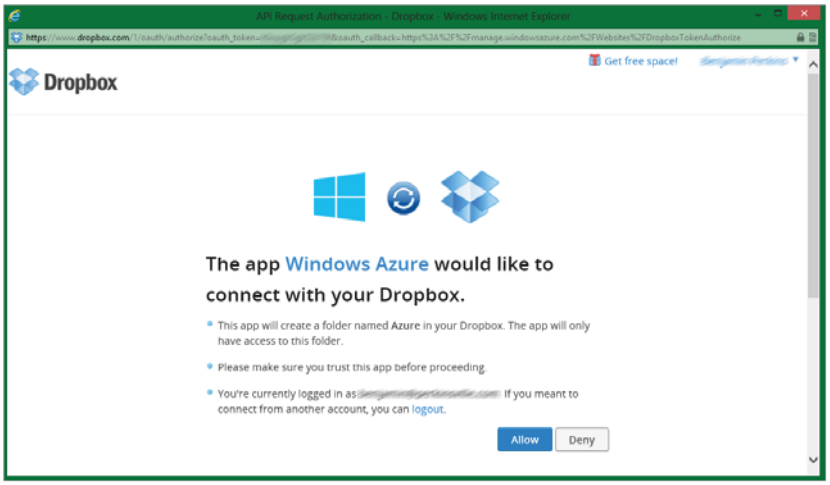

Dropbox
Get free space!
The app Windows Azure would like to connect with your Dropbox.
This app will create a folder named Azure in your Dropbox. The app will only have access to this folder.
Please make sure you trust this app before proceeding.
If you meant to connect from another account, you can logout.
Allow
Deny

FIGURE 5-13

Deploying Web Roles

The final deployment option discussed in this section is deploying a Web Role using a Service Definition File, Service Configuration File, and a Service Package, all of which are described in Table 5-4.

TABLE 5-4: Windows Azure Deployment Components

COMPONENT	DESCRIPTION
Service Definition File (.csdef)	Defines the service model and number of roles
Service Configuration File (.cscfg)	An XML based file the provides configuration settings for the Cloud Service, individual roles, and number of role instances
Service Package (.cspkg)	Contains application code and the Service Definition File

www.it-ebooks.info

You have numerous options for moving or replicating your source code to the Windows Azure platform. But remember that a Windows Azure Web Site and Windows Azure Web Role have different features. In addition, some deployment options do not provide source control, but rather only a source code depot for deployment. Depending on the complexity and requirements of your system, you need to assess which deployment option best meets your needs.

You can create both the deployment configuration and package files from Visual Studio 2012 after you install the Windows Azure SDK, using these basic steps:

1. Right-click the Web Role (not the project) and click Package, as shown in Figure 5-14. The process runs to create the required files for performing a package deployment.

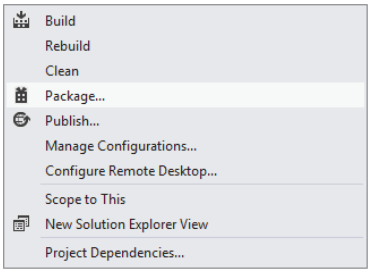


FIGURE 5-14

2. You deploy the created package using Visual Studio 2012, by accessing the Windows Azure Management Portal and selecting the Cloud Service to which you want to deploy the package.

NOTE *You can also create a deployment package using the CSPack command-line tool available from the Windows Azure SDK.*

---

3. Select either the New Production Deployment or New Staging Deployment options, as shown in Figure 5-15. A window appears enabling you to add the deployment package configuration files. If a deployment already exists, instead of seeing the New Staging Deployment option, you'll see the Update Staging Deployment option instead.

NOTE *It is recommended that you deploy the package to the Staging environment first, perform testing, and then, by selecting the Swap link, deploy to the Production environment.*

---

www.it-ebooks.info

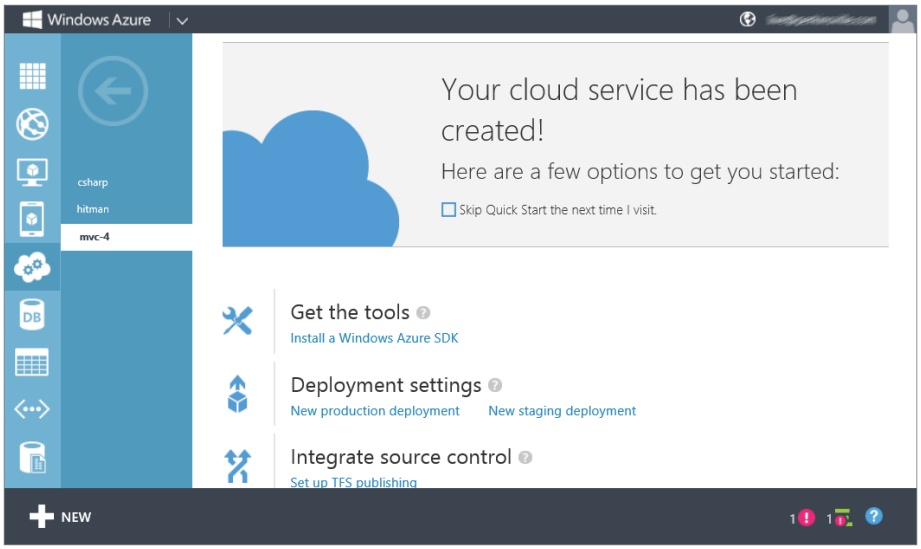


FIGURE 5-15

PLANNING DATABASE MIGRATION AND STORAGE

Most systems require a database to store and manage the capabilities or features that are available for users. The sample ASP.NET MVC 4 Web Role detailed in Chapter 6 requires a relational database that is accessed by NHibernate and uses the Model First approach, where you create the database structure by coding the class files required for implementation. With this approach, instead of the database driving the application, the design and relationship between the classes drives relationships already existing in the database.

However, many systems that you migrate to the Windows Azure platform usually have relational databases or data structures with a significant amount of data stored in them already. The question then becomes, "How do you migrate an existing database to the Windows Azure platform?" One answer is to use the SQL Service Migration Assistant, a free download from the Microsoft Download Center. You then use tool to convert an existing database to a SQL Server hosted on the Windows Azure platform. Numerous database-management system conversions are supported. For example, you can migrate Oracle, MySQL, Sybase, and Access databases to a Windows Azure SQL Server database using this tool.

NOTE *Database management, scaling, tuning, and maintenance are specialized skills. For further information about database options, review the "Choosing Your Windows Azure Services" section in this chapter, purchase a book specific to this topic, or review online documents that discuss this topic further.*

---

www.it-ebooks.info

After you create the structure and schemas of the database, it is recommended that you migrate the majority of your data using the SQL Server Integration Services (SSIS) available. Compared to importing the data directly into tables from a PC or other data source, migrating data using this tool can speed up the process and minimize incurred charges.

MONITORING THE STATUS OF A DEPLOYMENT

Watching the deployment of your code, knowing when it is complete, and monitoring any issues in its deployment are important pieces of information to capture, maintain, and communicate. For large-scale enterprise applications, this type of information is a critical step in the software design life cycle. You need to know that code has deployed to the staging or production environment before you can begin the testing phase, which in turn leads to making go/no-go decisions for a project.

A number of Azure features enable a Release Manager or deployment expert to observe code movement from the source code repository to either a Windows Azure Web Site or Web Role. When the code is ready for deployment, you can then move code to the Production or Staging Windows Azure environment. As discussed in the previous section, the Windows Azure Web Site supports the deployment of code from numerous sources, whereas the Web Role is currently bound to the Team Foundation Server.

The Windows Azure Web Site supports the most deployment methods:

- **FTP:** The most common deployment method for a hosting provider. After you create the Windows Azure Web Site, the FTP address appears in the Quick Glance column of the Dashboard for the website. With this address, you can use a tool, such as FileZilla, to deploy your code to the website, similar to when you publish to a hosting provider's server or to a standalone server. Most FTP tools have a status bar to show how far along you deployment is.
- **Dropbox:** If you deploy your source code using Dropbox, you can view your deployment status in the Windows Azure management console. As illustrated by Figure 5-16, the status tracks deployment from the moment you make a connection between the Windows Azure Web Site and Dropbox.

During deployment, the status of synchronization updates as the static files and binaries deploy. When complete, you can test to determine if the code has installed as expected.

Deployment history also provides a list of deployments, containing the number of items changed, identification of which one is the most current and the date when the deployment was made (see Figure 5-17). Deployment history is available when you deploy from a number of repositories — not only Dropbox. For example TFS and GitHub both provide history, while FTP does not. Git and CodePlex have the same features as Dropbox.

www.it-ebooks.info

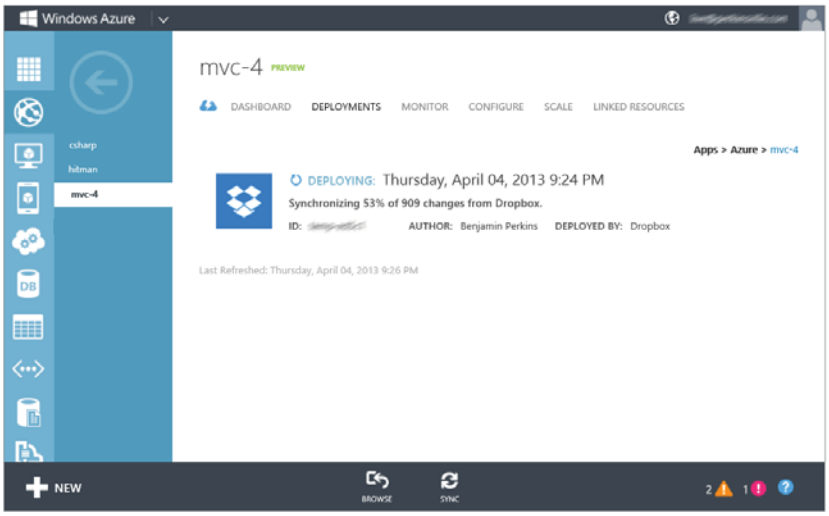


FIGURE 5-16

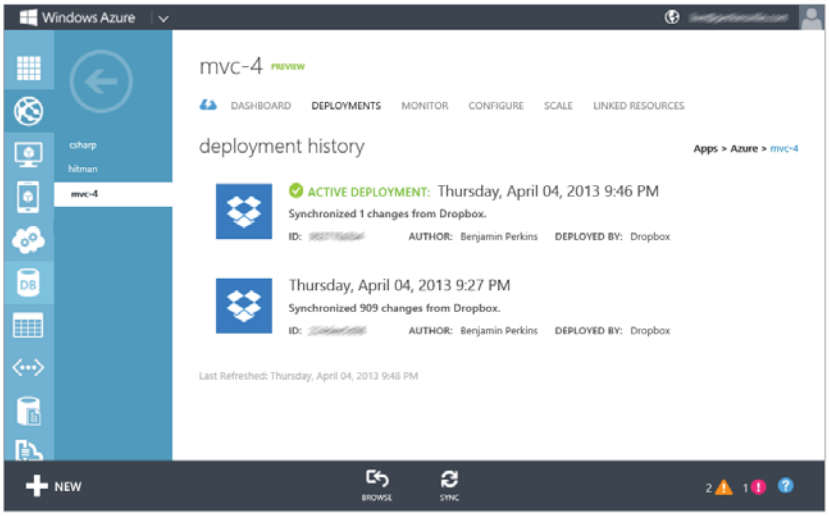


FIGURE 5-17

In addition to these deployment options, Team Foundation Server (TFS) is available for both the Windows Azure Web Site and Web Role. When you configure the link between TFS and your Windows Azure Web Site or Web Role deployment occurs immediately after you check your code into the TFS source code repository. For example, in the Visual Studio 2012 IDE (which is connected to the same TFS instance bound to the Windows Azure Web Site or Web Role), when you make a code change and check in, shown in Figure 5-18, these changes automatically deploy to either your website or Web Role.

www.it-ebooks.info

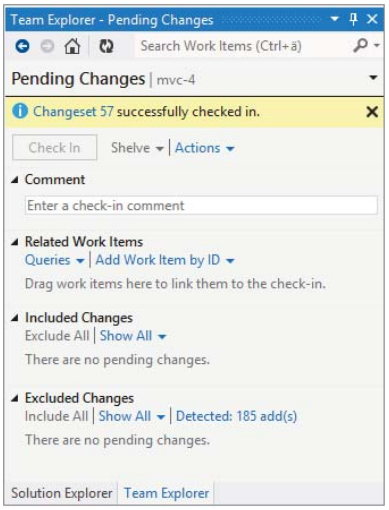

Team Explorer - Pending Changes
Search Work Items (Ctrl+ä)
Pending Changes | mvc-4
Changeset 57 successfully checked in.
Check In
Shelve | Actions
Comment
Enter a check-in comment
Related Work Items
Queries | Add Work Item by ID
Drag work items here to link them to the check-in.
Included Changes
Exclude All | Show All
There are no pending changes.
Excluded Changes
Include All | Show All | Detected: 185 add(s)
There are no pending changes.
Solution Explorer
Team Explorer

FIGURE 5-18

To learn about the many features available for monitoring and managing deployments, you can use the online version of TFS via your Microsoft ID at `http://***.visualstudio.com`, where *** is the name provided to you when you created your TFS connection with Windows Azure. For example, for a given project, a complete list of currently queued builds, completed builds, and deployed builds are available for real-time and post-mortem analyses.

SUMMARY

In this chapter you learned many decision points, capabilities, and limitations for deployment to the Windows Azure platform. You learned under which circumstances certain websites or applications should not be deployed to Windows Azure, primarily based on the incurred cost of the platform. A discussion occurred for SDKs supported on the Windows Azure platform such as PHP, Python, Java, and .NET.

The different Cloud Computing Services such as IaaS, PaaS, and SaaS were discussed as well as the primary differences between a Windows Azure Web Site and Web Role, which is also commonly referred to as a Cloud Service. Data storage options, such as Table Storage and SQL Server database were discussed in addition to how to deploy to them and how to deploy to the website or Web Role using source code repositories such as Git, TFS, Dropbox, or CodePlex. There are also many options for monitoring the deployments in real-time so that testing can begin when complete.

In the following chapter, you can find exercises and examples using many of the concepts discussed in this chapter.

www.it-ebooks.info

6 Deploying an ASP.NET MVC 4 Project to Windows Azure

EXERCISES AND EXAMPLES

IN THIS CHAPTER

- How to access Windows Azure
- How to create the Window Azure Web Site and Cloud Service
- How to add and connect a SQL database
- How to deploy and test your code
- Establishing a Team Foundation Server (TFS) connection and publishing the ASP.NET MVC 4 website
- How to connect a Windows Azure Web Site to a GitHub code repository
- Using FTP to publish an ASP.NET MVC 4 website

WROX.COM CODE DOWNLOADS FOR THIS CHAPTER

You can find the wrox.com code downloads for this chapter at `www.wrox.com/go/azureaspmvcmigration` on the Download Code tab. It is recommended that you download the code and review it so that you have a good understanding of what is about to be discussed.

Now that you understand many of the deployment options available on the Windows Azure platform, it is time to put your recently learned knowledge to work. By deploying source code and data structures to a Web Site or Cloud Service hosted on Windows Azure, users, customers, or visitors can access and use its features. If you have performed the exercises described in this book up to this point, you've created a website and a Cloud Service; however, they exist only on your development PC and only you can access them. For other people to view the site, you must deploy it to a server that's publicly accessible. Once you complete this chapter, you'll have a deeper understanding of what the different types of Windows Azure platform deployment services offer and your website or Cloud Service will be available for anyone to access.

www.it-ebooks.info

NOTE *The following exercises provide step-by-step instructions for making different types of deployments. If you're not familiar with the concepts behind this chapter, Chapter 5, discusses the various concepts behind deploying an ASP.NET project to Azure.*

ACCESSING WINDOWS AZURE

Before you can perform a deployment to the Windows Azure platform, you need to create an account and choose the kind of subscription you need. A Windows Azure account is what resource usage is reported against and is the entity in which the services are billed to. The subscription describes the Windows Azure services to which the account has access.

To access the Windows Azure platform, you need a Microsoft ID; this was formally called a Windows Live ID. If you currently have either types of ID, including an Office 365 ID, you can bind any of them to your Windows Azure account when you create it.

To view the different subscription types, follow these steps:

1. Open a web browser and navigate to `www.windowsazure.com`.
2. Click the Pricing link at the top of the page, as shown in Figure 6-1. You are now navigated to the Calculator that enables you to choose between numerous Windows Azure services and configure them to see what the cost would be.

NOTE *The pace of change on the Azure platform is happening fast. Figure 6-1 might change at some point. If this happens, search for the pricing information on the website mentioned in Step 1.*

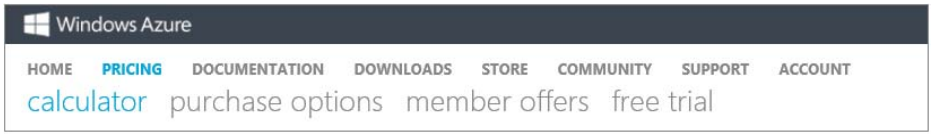


FIGURE 6-1

3. Navigate to the Purchase Options to pick a subscription that best meets your requirements. Subscription details are likely to change over time due to the very competitive state of the cloud and the current players such as Google, Amazon.com, Sales Force, and Microsoft. For example, MSDN subscribers have a monthly credit available to use against their Azure charges. This is something newly provided at the time of writing this book. At the moment, there are three subscription options: pay as you go, 6-month offer, and 12-month offer. Each has its own benefits, so research and choose which option best meets you needs.

www.it-ebooks.info

4. Regardless of which option you choose, when selected, you are walked through an Account Creation Wizard that collects details and required information for the Windows Azure account creation.
5. When completed, navigate to `http://manage.windowsazure.com`, enter your credentials, and complete the following exercises.

If you want to test only the Windows Azure platform, you can use the 90-day free trial before you purchase a subscription. This is a great option and you get 90 days to create websites, Cloud Services, virtual machines, SQL databases, and so on and see how they all work together.

If you have an MSDN subscription, you can get some free benefits based on your MSDN subscription level, for example (at the time of writing this chapter):

- 375–1,500 hours of the small compute instance
- 500,000–2,000,000 CDN transaction
- 70 GB–90 GB of storage
- 50,000,000–100,000,000 storage transactions
- 1–5 SQL Database Web Edition
- 30 GB–40 GB of data transfers

If you are an individual who wants to move your websites to the Windows Azure platform, seriously consider purchasing an MSDN subscription and use the free Windows Azure benefits. You will receive a lot of other benefits in addition.

CREATING THE WINDOW AZURE WEB SITE AND CLOUD SERVICE

Now that you have access to the Windows Azure platform, you can create Web Sites, Cloud Services, and a SQL database. First, you should create a website instance that you can later use to host the ASP.NET MVC 4 project you created in Chapter 2, and fine-tuned in Chapter 4. After the creation of the website, I'll discuss the creation of the Cloud Service. The Cloud Service, which is synonymous with Web Role, will host the same ASP.NET MVC 4 project after a simple conversion from a project to a Web Role, also discussed in this chapter.

Creating a Website

To create a website on the Windows Azure platform, follow these steps:

1. Click +NEW located on the bottom left of the page, as shown in Figure 6-2. When selected a pop-up window is rendered, as shown in Figure 6-3.

www.it-ebooks.info

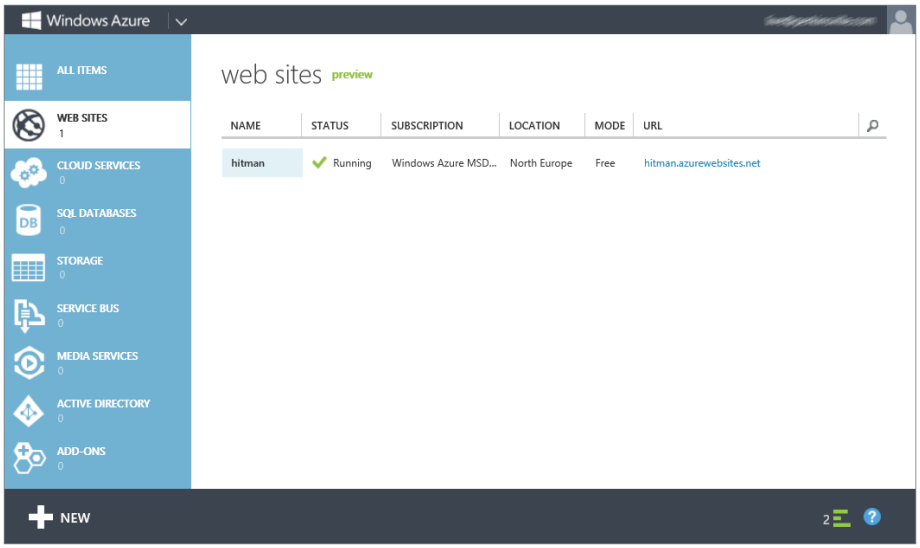


FIGURE 6-2

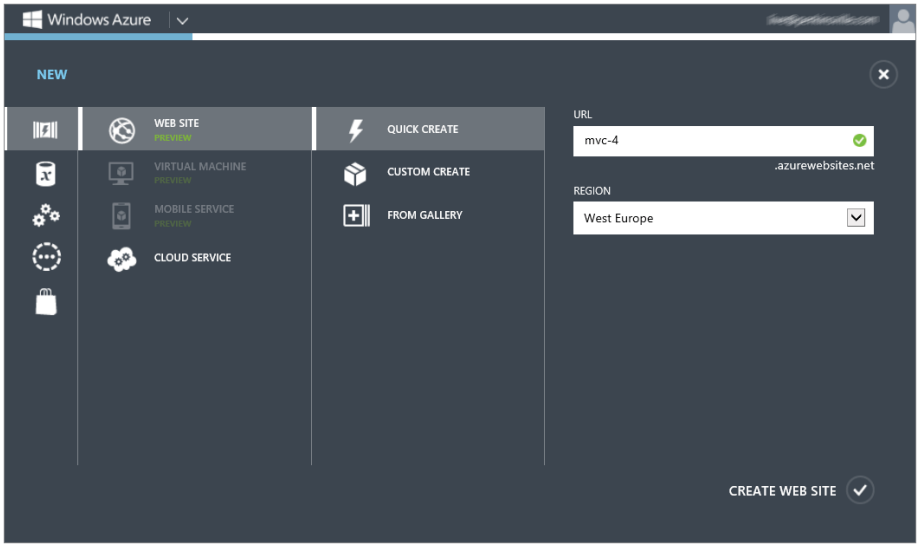


FIGURE 6-3

2. You are presented three different options for creating a website:

   - **Quick Create:** Enables you to quickly create a website by specifying the URL and Region only. You can perform other configurations after creating the website.

www.it-ebooks.info

- **Custom Create:** In addition to specifying the URL and Region, this option enables you to choose a database that the website connects to, and a source for publishing your source from. Currently, Team Foundation Server and Git are supported.
- **From Gallery:** When this item is selected, it enables you to install a number of templates to use with your website, for example, WordPress, Das Blog, or .DotNetNuke.

For this example, select Quick Create.

3. Enter an available URL and the Region, and select the Create Website link on the lower right.
4. When created, click the Website and view the Website Dashboard, as illustrated in Figure 6-4. If the QuickStart page come up first, select the DASHBOARD link located at the top left of the main page.

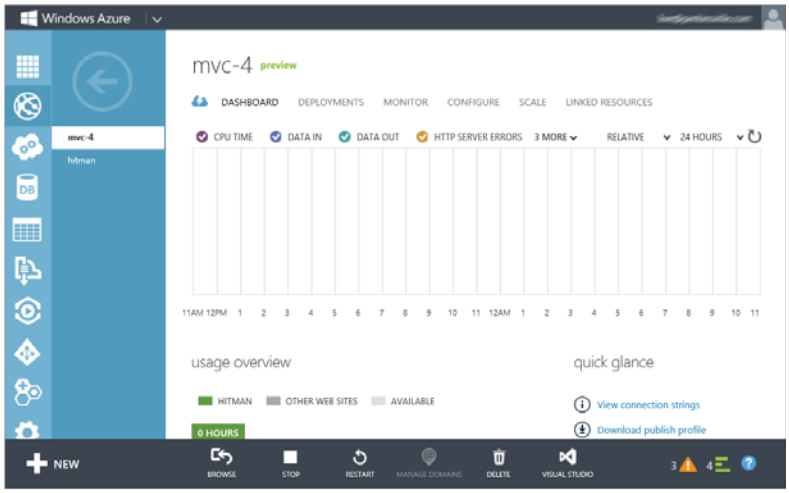


FIGURE 6-4

The website on the Windows Azure platform is now created and ready for you to publish some source code. If you would like to jump straight to the deployment exercise, skip the next section, "Creating a Cloud Service," and go directly to the "Adding a SQL Server Database" section. Come back to the "Creating a Cloud Service" after you are ready to create the Cloud Service, and migrate the ASP.NET MVC 4 project to a Web Role.

Creating a Cloud Service

A Cloud Service falls into the category of a PaaS, meaning, the administrator does not need to be concerned about the operating system, hardware, and network components of the solution.

www.it-ebooks.info

In addition, a Cloud Service can be connected to via a Remote Desktop Connection and administrated similar to a normal server. To create a Cloud Service, follow these steps:

1. In the Windows Azure Management console, click the Cloud Services menu item; then click the +NEW link on the bottom left of the web page, as shown in Figure 6-5. When clicked a pop-up is rendered that walks you through the creation of the Cloud Service. Figure 6-6 illustrates how this pop-up looks.

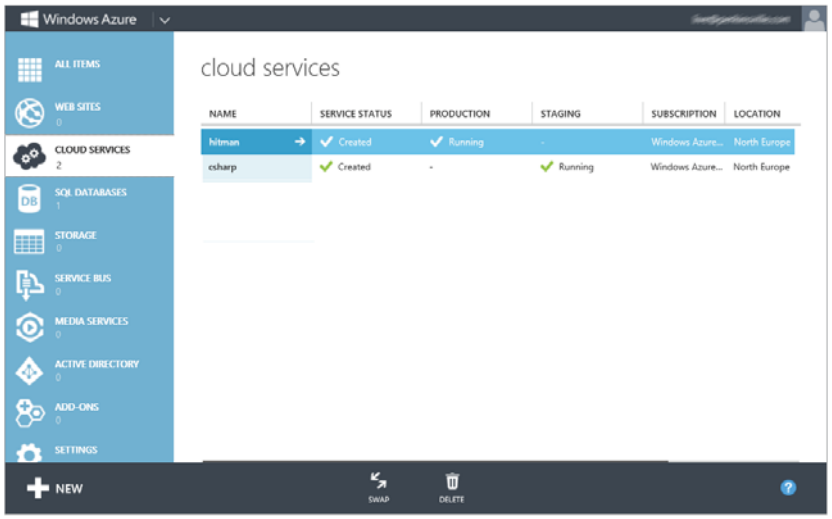


FIGURE 6-5

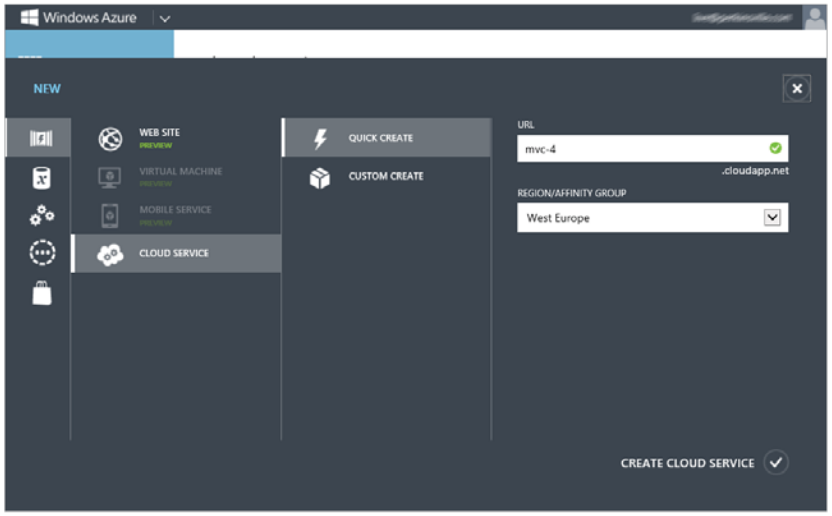


FIGURE 6-6

www.it-ebooks.info

2. Select the Quick Create menu item, and enter the URL and Region/Affinity Group you would like for your Cloud Service.
3. When complete, select the Create Cloud Service link on the lower right of the pop-up window.

After clicking the Create Cloud Service Link, you are redirected to the list of Cloud Services you have in your Windows Azure Subscription. Wait until the status shows as completed, then you can click the Cloud Service and view its details.

ADDING AND CONNECTING A SQL DATABASE

Now that you have access to the Windows Azure platform, and have created the Windows Azure Web Site or a Cloud Service, the next step is to add the SQL database. The sample ASP.NET MVC 4 project uses the database to retrieve blog and comment information.

To add a SQL database to the Windows Azure platform, the basics steps involve logging into Windows Azure and selecting the SQL Databases menu item; creating a new SQL Database if this is the first database you're created; providing details such as Login Name, Password, as so on. After you do all this, you can view the details found on the database Dashboard.

Adding a SQL Server Database

To create the SQL Database on Windows Azure, follow these steps:

1. Select the SQL Databases menu item on the left navigation pane. Assuming that you do not have exiting databases, select the Create a SQL Database link on the feature window, as shown in Figure 6-7.

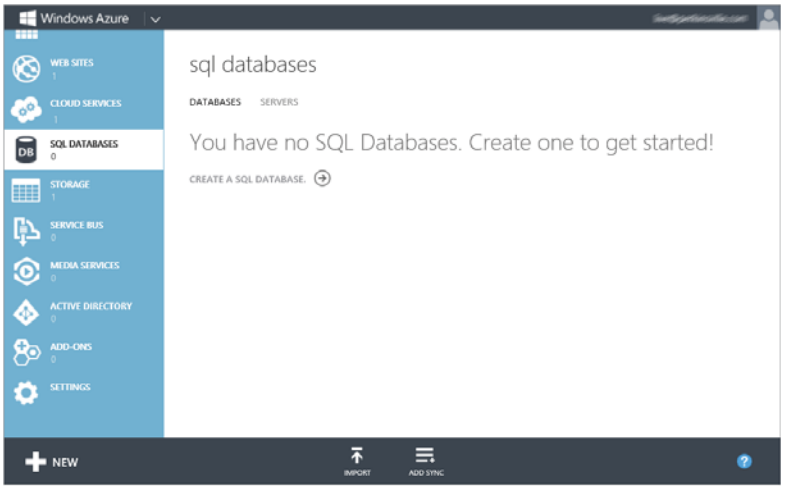


FIGURE 6-7

www.it-ebooks.info

2. In the Specify Database Settings window that appears (see Figure 6-8), enter **Blogs** as the database Name, and select Web as the database Edition. Keep the default Collation name. Finally, select the New SQL Database Server from the drop-down, and then click the arrow on the bottom right to go to the next step. Figure 6-9 displays the final request for information before the database is created.

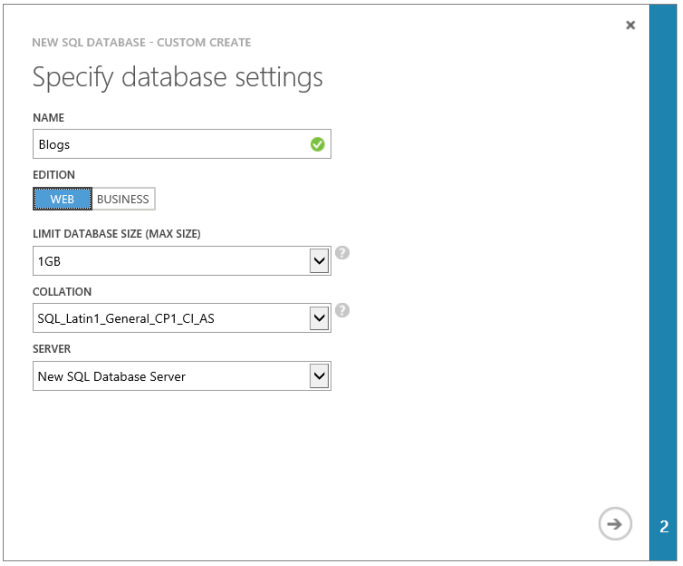


FIGURE 6-8

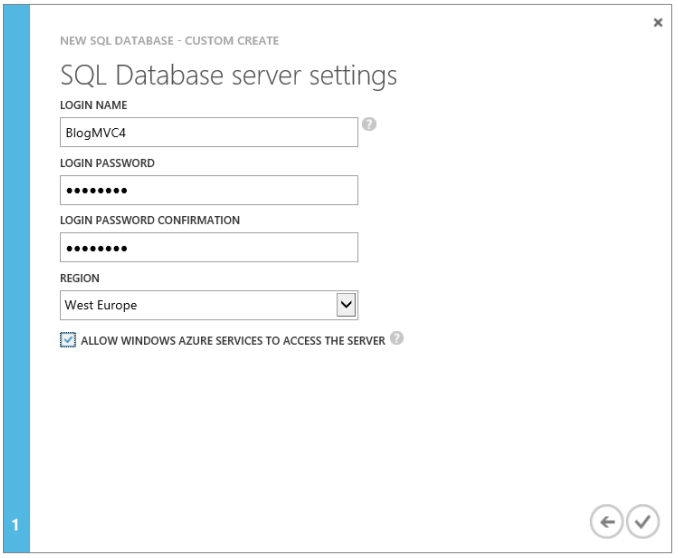


FIGURE 6-9

www.it-ebooks.info

NOTE *The difference between the Web and Business edition has to do with capacity and the billing model. For example, the Web edition scales from 1 GB to 5 GB, whereas the Business edition scales from 100 GB to 150 GB. Because this is just a test example, choose Web ➪ Server ➪ Create SQL Database. Also note that collation has to do with supporting, sorting, and comparing certain character types. So unless you have some special needs, leave this as the default setting. Note that you cannot change this parameter after you create the database.*

3. In the SQL Database server settings, enter and remember the database Login Name and Password. Select the Region that is closest to your customers, and select the check button on the bottom right to create the database.

   If this is not the first time you are creating a database in your Windows Azure environment, the Create New SQL Database does not appear in the main section. Instead, a list of the existing databases display.

4. To add a new database, select the +NEW button on the lower-left side of the page previously shown in Figure 6-7. When selected, the pop-up, as shown in Figure 6-10, appears.

5. Select the Quick Create item, enter the requested information (Database Name and Server location), and select the Create SQL Database button.

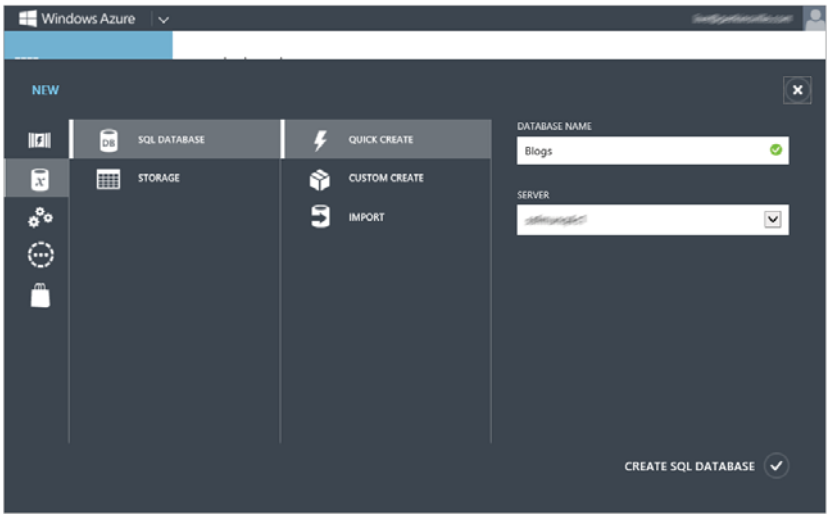


FIGURE 6-10

www.it-ebooks.info

6. Select Quick Create for creating an SQL database in this exercise. Notice that you have three options:
   - **Quick Create:** A good option for creating a simple database and then making the configurations later via the database Dashboard. You can choose Database Name and Server using this option.
   - **Custom Create:** Provides a few more options during the creation of the SQL Database. You can choose Database Name, decide between Web and Business edition, set the max size of the database, choose the collation method, and then select the server where the database is stored.
   - **Import:** Enables you to import a saved database from your BLOB storage account.
7. After the database is created and has a status of Online, click the Name, and you are redirected to the Dashboard, as shown in Figure 6-11. You can find some important information on this screen, for example, the usage statistics, the connection string that applications can use to connect to the database, troubleshooting tips, back-up and restore capabilities, and a URL to remotely connect to the database.

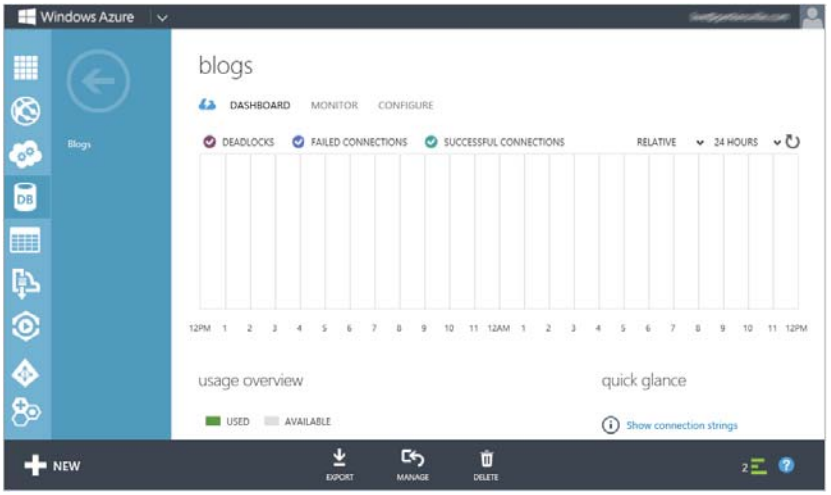


FIGURE 6-11

Connecting to the Database

You can add a connection to this database from Visual Studio. To add an additional layer of security, you're required to configure the database on Widows Azure to allow access based on the IP address of the client. To do this, follow these steps:

www.it-ebooks.info

1. In the Dashboard for the Blogs database, look on the right side for a section titled Quick Glance. In that list of links, find and select the one titled Manage Allowed IP Addresses. By default, the page shows the IP address of the machine currently connected to the Windows Azure platform, as shown in Figure 6-12.
2. Select the Add to Allowed IP Addresses link next to the IP address, and then click the Save button at the bottom of the page to allow the machine with that IP address to connect to the database remotely.

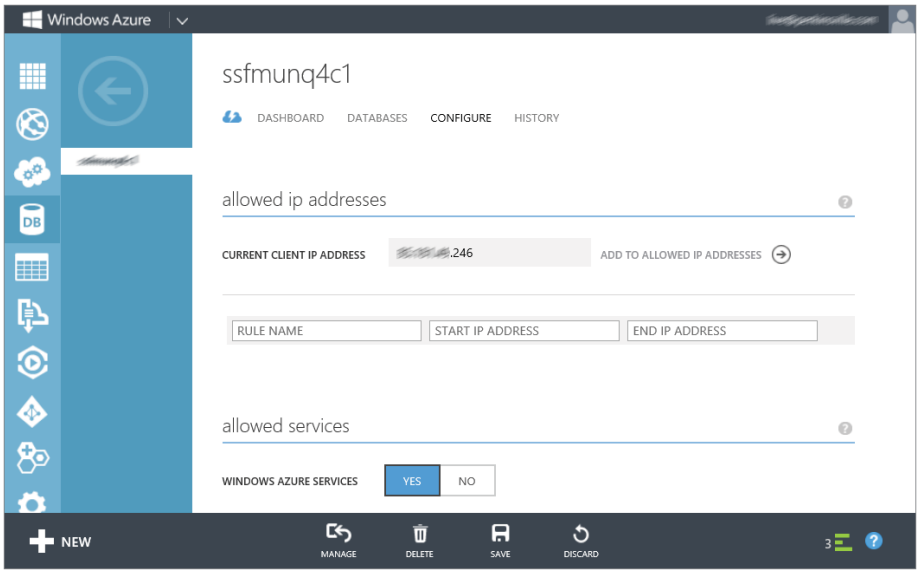


FIGURE 6-12

3. To add the connection from within the Visual Studio 2012, open the Server Explorer, right-click Data Connection, and then select Add Connection, as shown in Figure 6-13.

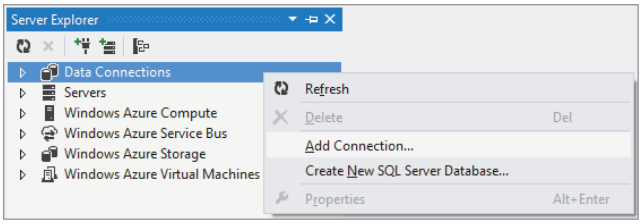


FIGURE 6-13

www.it-ebooks.info

4. In the Server Explorer window that appears, you can find the server name where your database is created on the right side of the database Dashboard window under the label Server Name.

5. Enter that server name into the Add Connection window, select the Use SQL Server Authentication option, and enter the Login Name and Password that you created in Figure 6-3 using the steps in the section "Creating a Website." After clicking the Select or Enter a Database Name option, select the Blogs database, and then click the Test Connection button. If successful, you see the Test Connection Succeeded message, as shown in Figure 6-14.

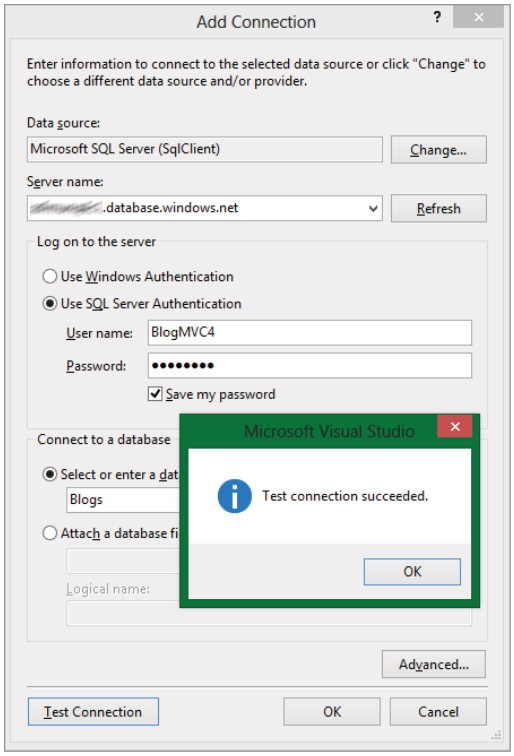


FIGURE 6-14

6. In Visual Studio, you can right-click the database and select Properties. In the Properties window on the lower right of the Visual Studio IDE, look in the Connection String element. You can use this value from in your Windows Azure-hosted application to connect to the database.

You now have successfully created a SQL Server database on the Windows Azure platform and have the connection string required to connect to it. You can now move onto the next section and deploy the code that uses this database.

www.it-ebooks.info

DEPLOYING AND TESTING YOUR CODE

Now that you have a Windows Azure account, your Website and/or Cloud Service, and a SQL database, the next step is to deploy the ASP.NET MVC 4 source code and the dependencies to the cloud so that user, customers and visitors can access it. There are two kinds of Windows Azure features that support websites. As previously discussed in Chapter 5, they are Web Sites and Cloud Services.

The sample ASP.NET MVC 4 project for this chapter was created in a way that restricts its deployment to a website. Therefore, the section "Converting an ASP.NET MVC 4 Project to a Cloud Service" describes the steps required to transition the ASP.NET MVC 4 project from a website to a Cloud Service. However, you must first make a few modifications that apply to both the Web Site and Cloud Service project before migrating them to your Windows Azure instances. They are as follows:

1. **Change the path to the XML RSS blog list.** In Chapter 2 the path to the XML RSS file was hard-coded, as shown in the following code snippet:

   ```
   XDocument doc = XDocument.Load(@"C:\...\MVC\MVC\Content\RSS\csharp2011.xml");
   ```

   You must update this to use a relative path. Update this line of code located in the `Controller\HomeController.cs` and `Controller\BlogController.cs`, as shown in the following code snippet:

   ```
   XDocument doc = XDocument.Load(Server.MapPath("Content/RSS/csharp2011.xml"));
   ```

2. **Change the Connection String to point to the previously created SQL database.** In the previous section, you added the connection to the database to the Server Explorer Data Connections. Go back to this, right-click the database connection, and select Properties. Look in the Properties window to find the value located in the Connection String attribute. Copy this value. (Highlight it and press Ctrl+C.) Then open the `ConfigureNHibernate()` method located in the Global.asax.cs file, and modify the `db.ConnectionString` property to be that of the Windows Azure SQL database. For example, the existing value is as shown in the following code snippet:

   ```
   db.ConnectionString =
         "Data Source=.\SQLExpress;
          Initial Catalog=Blogs;
          Persist Security Info=True;
          User ID=sa;Password=****;Pooling=False";
   ```

   This code snippet makes a connection to the SQL Server Express instance on a development PC that was installed and configured in Chapter 2. Change this value to your instance of the SQL database created on Windows Azure. The following code snippet is an example of how it may look:

   ```
   db.ConnectionString =
         "Data Source=SERVERNAME.database.windows.net;
         Initial Catalog=Blogs;
   ```


www.it-ebooks.info

```
        Persist Security Info=True;
        User ID=BlogMVC4;Password=********";
```

3. **Uncomment, execute, and comment out the logic that creates the database tables using NHibernate methods.** Recall from Chapter 2 the segment of code that creates the database schema and populates the database with data. This code segment is located in the `Application_Start()` method and shown in the following segment shown in Listing 6-1.

LISTING 6-1: NHibernate Create Database Code Segment

```
new SchemaExport(configuration).Drop(false, true);
new SchemaExport(configuration).Create(false, true);
SchemaValidator schemaValidator = new SchemaValidator(configuration);
schemaValidator.Validate();
InsertTestData();
```

NOTE *Be sure to comment this with your first deployment; then, after the database is successfully created, comment out this code segment and republish the file.*

Converting an ASP.NET MVC 4 Project to a Cloud Service

As previously discussed, there are numerous realized benefits from using a Cloud Service versus a website. The following details cover the steps required to convert the ASP.NET MVC 4 website to a Cloud Service. Perform the following to make the conversion:

1. **Download and install the Windows Azure SDK for .NET.** To install the Windows Azure SDK for .NET, open Visual Studio 2012 and select File ➪ New ➪ Project. In the Template folder select Cloud. If you have not already manually installed the SDK, you will see that the Get Windows Azure SDK for .NET is the only selectable choice provided.
2. **Select Get Windows Azure SDK for .NET, and click OK.** Figure 6-15 illustrates the New Project window.
3. **Download the SDK.** Follow the Installation Wizard (shown in Figure 6-16), which includes the download and installation of the Web Platform Installer and completes the SDK installation. You will need to close any open instances of Visual Studio.
4. **Create the ASP.NET MVC 4 Web Role.** After the installation is complete, open Visual Studio 2012 in elevated mode (that is, run as Administrator). Select File ➪ New ➪ Project.

www.it-ebooks.info

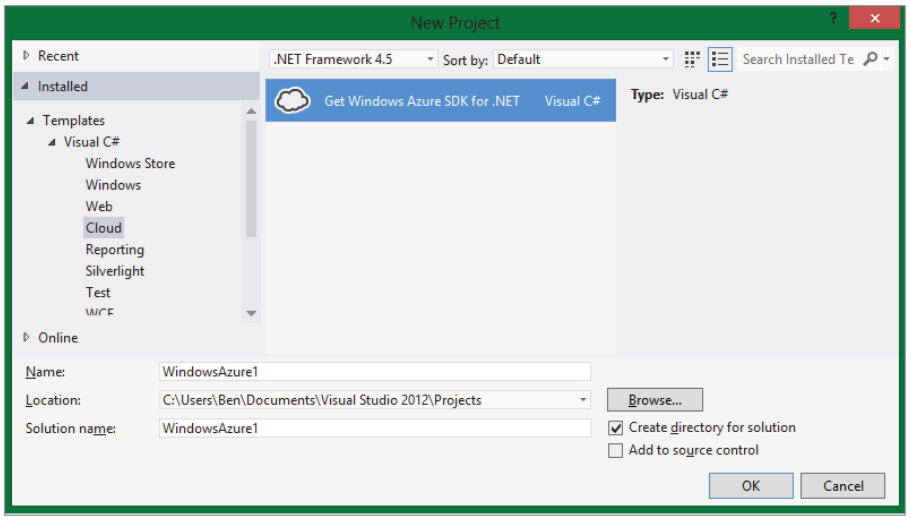


FIGURE 6-15

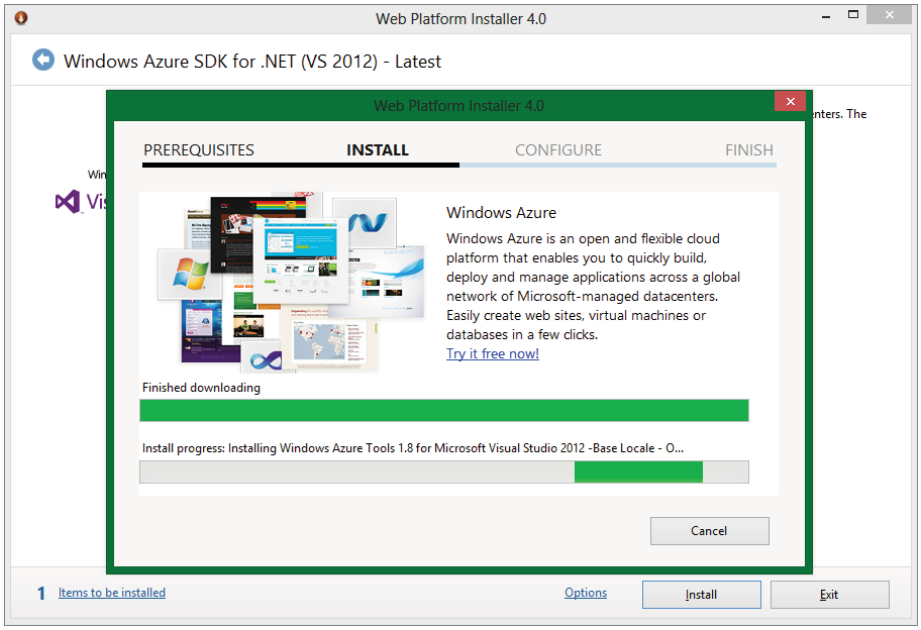

Web Platform Installer 4.0
Windows Azure SDK for .NET (VS 2012) - Latest
Web Platform Installer 4.0
PREREQUISITES
INSTALL
CONFIGURE
FINISH
Windows Azure
Windows Azure is an open and flexible cloud platform that enables you to quickly build, deploy and manage applications across a global network of Microsoft-managed datacenters. Easily create web sites, virtual machines or databases in a few clicks.
Try it free now!
Finished downloading
Install progress: Installing Windows Azure Tools 1.8 for Microsoft Visual Studio 2012 -Base Locale - O...
Cancel
1 Items to be installed
Options
Install
Exit

FIGURE 6-16

5. **Create a name for the Web Role.** Select the Cloud Template ➪ Windows Azure Cloud Service, and enter the name **WindowsAzureMVC**. Select OK. Figure 6-17 shows the Add Project window. The New Windows Azure Cloud Service window opens, as shown in Figure 6-18.

www.it-ebooks.info

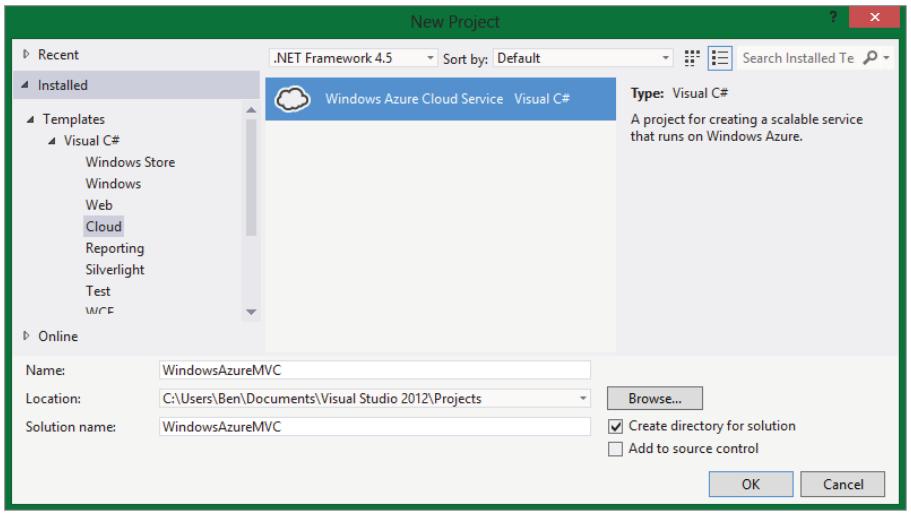


FIGURE 6-17

6. **Define the Web Roles.** Select ASP.NET MVC 4 Web Role; then click the ">" button to add the Web Role to the solution. Hover over the just-added Web Role, and click the Edit button. Rename the Web Role to any name you would like or leave it as is; then click OK. (**csharp** is used in this example.) In the next window, select Internet Application, leave all of the other default settings, and click OK.

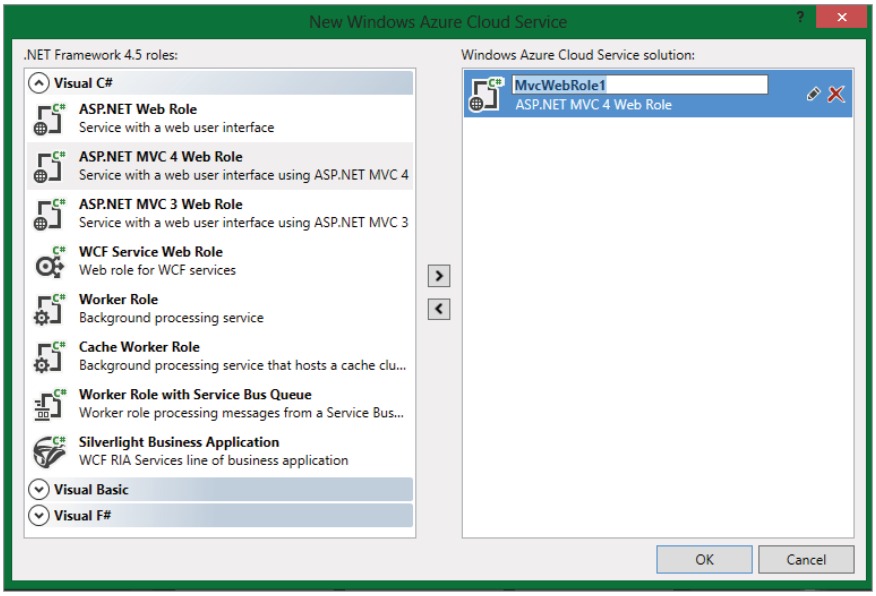

New Windows Azure Cloud Service
.NET Framework 4.5 roles:
Visual C#
ASP.NET Web Role
Service with a web user interface
ASP.NET MVC 4 Web Role
Service with a web user interface using ASP.NET MVC 4
ASP.NET MVC 3 Web Role
Service with a web user interface using ASP.NET MVC 3
WCF Service Web Role
Web role for WCF services
Worker Role
Background processing service
Cache Worker Role
Background processing service that hosts a cache clu...
Worker Role with Service Bus Queue
Worker role processing messages from a Service Bus...
Silverlight Business Application
WCF RIA Services line of business application
Visual Basic
Visual F#
Windows Azure Cloud Service solution:
MvcWebRole1
ASP.NET MVC 4 Web Role
OK
Cancel

FIGURE 6-18

www.it-ebooks.info

7. **Add the NHibernate package using the Package Manager and NuGet.** When the project creation process successfully completes, select Tools ➪ Library Package Manager ➪ Package Manager Console, and enter `install-package NHibernate` to install the NHibernate binaries. Figure 6-19 displays the outcome of the execution of that command.

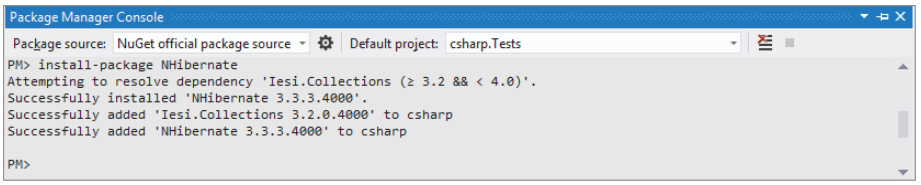


FIGURE 6-19

The command `install-package NHibernate` automatically adds the required DLL libraries to the References directory of the solution. The NHibernate components can now be referenced within the source code so that its features can be used.

Adding ASP.NET MVC 4 Website Code to the ASP.NET MVC 4 Web Role/Cloud Service

To complete the migration from the ASP.NET MVC 4 website to a Web Role so that you can deploy it to the previously created Windows Azure Cloud Service instance, complete the following:

1. **Open two different instances of Windows Explorer.** You'll open one for the ASP.NET MVC 4 website and another for the ASP.NET MVC 4 Web Role.
2. **Navigate to the locations where you stored both of these projects and open them.** This is shown in Figure 6-20.

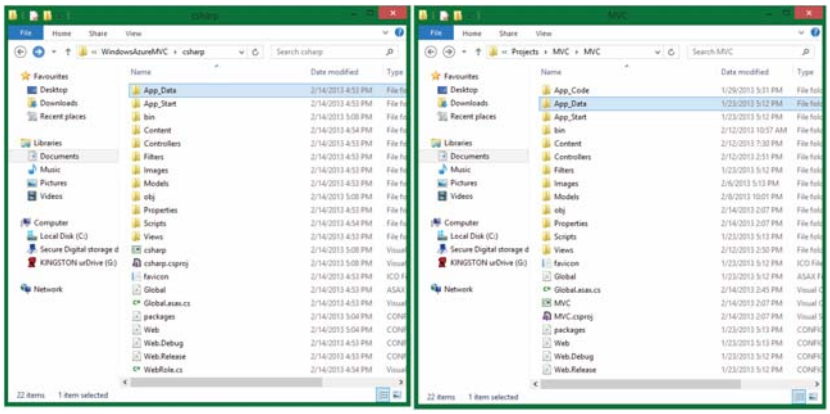


FIGURE 6-20

www.it-ebooks.info

3. **Copy the following files from the MVC project to the WindowsAzureMVC project:**
   - MVC\Content\RSS directory and contents
   - MVC\Content\Syntax directory and contents
   - MVC\Content\Site.css
   - MVC\Controllers\BlogsController.cs
   - MVC\Controllers\CSharpFundamentalsController.cs
   - MVC\Controllers\Help.cs
   - MVC\Controllers\HomeController.cs
   - MVC\Controllers\LessonsController.cs
   - MVC\Controllers\NewsController.cs
   - MVC\Controllers\ReviewsController.cs
   - MVC\Images all contents
   - MVC\Models\Blog.cs
   - MVC\Models\BlogArchives.cs
   - MVC\Models\BlogList.cs
   - MVC\Models\BlogNavBarElements.cs
   - MVC\Models\Comments.cs
   - MVC\Views\Blogs directory and contents
   - MVC\Views\CSharpFundamentals directory and contents
   - MVC\Views\Help directory and contents
   - MVC\Views\Home directory and contents
   - MVC\Views\Lessons directory and contents
   - MVC\Views\News directory and contents
   - MVC\Views\Reviews directory and contents
   - MVC\Views\Shared directory and contents
   - MVC\Favicon.ico
   - MVC\Global.asax.cs

4. **Add each of the copied files to the WindowsAzureMVC project.** From within Visual Studio 2012, right-click the Content directory ➪ Add ➪ New Folder ➪ RSS. Then

www.it-ebooks.info

right-click the RSS folder ➪ Add ➪ Existing Item. Navigate to the WindowsAzureMVC\Content\RSS directory, and select the CSHARP2011.xml file.

5. Perform the action in step 4 for all the files identified in step 3.

NOTE *You must change all places that reference the* `MVC` *namespace, for example using* `MVC.Models`*, to the name of the Web Role you selected, (such as* `csharp`*), as shown previously in 6-18. Search in all the model, view, and controller files for this entry. If you get lost, you can download the source code as an example, from the Wrox website, to help you get through any difficulty.*

---

6. When all the changes are complete, select F5 to locally run the Web Role, and click around the website making sure everything is working as expected.

After you are certain that it is ready to be published, continue to the next section, "Deploying to a Windows Azure Cloud Service," and follow those instructions to move the Web Role to the Windows Azure Cloud Service.

Deploying with Visual Studio Publishing Features

As you know, there are two different types of hosting capabilities on the Windows Azure platform: the website and the Cloud Service. At this point you now have two ASP.NET MVC 4 projects. One is a Windows Azure Web Site created in Chapter 2, and the other is a Windows Azure Cloud Service migrated from the website in the previous section.

Use the instructions in the following two sections to deploy your website or Cloud Service to the Windows Azure platform.

Deploying to a Windows Azure Web Site

After making changes to the Connection String and changing the path to the XML RSS feeds used for presenting the most current blogs on the Home and Blogs page, the ASP.NET MVC 4 project is ready for deployment to the previously created Windows Azure Web Site.

To publish the ASP.NET MVC 4 website to the Windows Azure Web Site, complete the following steps:

1. **Create a publish profile name.** To create the publish profile, right-click the project name, and select Publish, as shown in Figure 6-21. From the drop-down box on the Publish Web window, select <New...> and enter the profile name, as illustrated with Figure 6-22.

www.it-ebooks.info

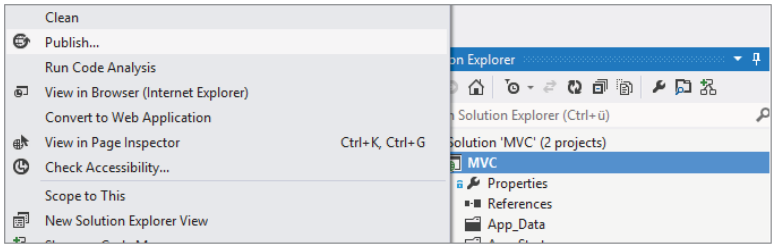


FIGURE 6-21

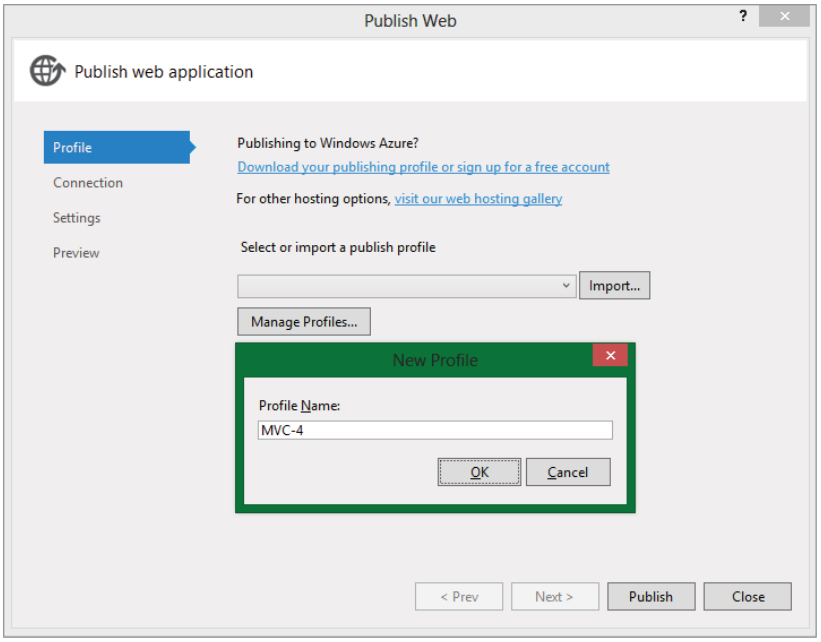


FIGURE 6-22

NOTE *Creating a publish profile lets you store connection information and settings specific to this project. If, in the future, you have more than a single project hosted on Windows Azure, you can select the profile for the project and publishing happens without the need to reconfigure the settings.*

2. **Create the connection to the Windows Azure Web Site.** When all of the information is entered, select the Validate Connection to confirm that the provided information is valid. Figure 6-23 demonstrates this activity. Select the Next button.

www.it-ebooks.info

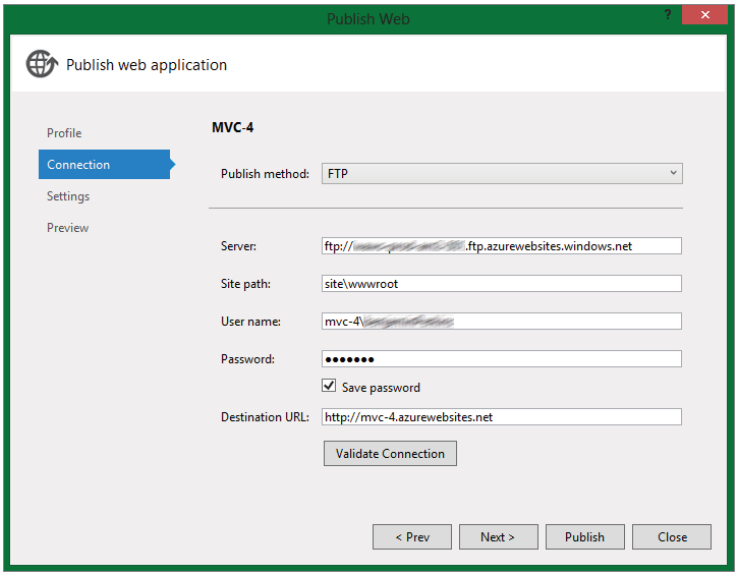


FIGURE 6-23

3. **Select publishing settings.** Enter the settings, as shown in Figure 6-24. In this example, select the Delete All Existing Files Prior to Publish check box. This is done to make sure that files you no longer need are removed to avoid conflict with other files, in the event that the versions of dependent files get out of sync. You can also choose to compile the code during deployment and to exclude the file from the App_Data folder.

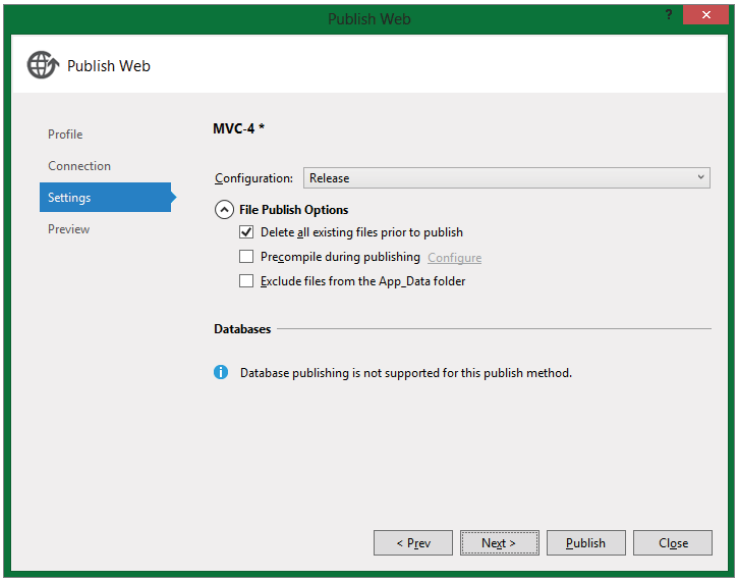


FIGURE 6-24

www.it-ebooks.info

4. **Select the Configuration type (Release or Debug)**. Select Release from the Configuration drop-down; however, if you were publishing to a Staging instance on your Cloud Service you might consider or need to deploy the Debug project.

5. **Publish the ASP.NET MVC 4 website.** Select the Next button to preview the configuration; then select the Publish button. During the publishing you should see the status as shown in Figure 6-25.

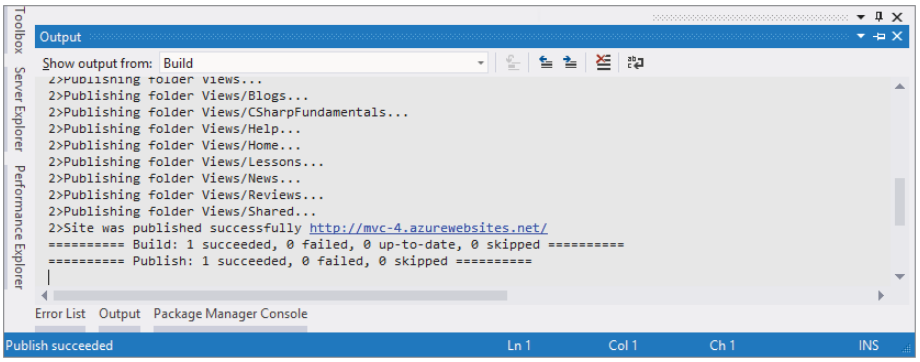


FIGURE 6-25

6. **Review the result of the published website.** After the publishing is complete, the website address entered into the Destinations URL, as shown in Figure 6-23, opens.

WARNING *After the website has been run once, don't forget to comment out the code that creates the database structure from the Global.asax.cs file and then republish.*

Deploying to a Windows Azure Cloud Service

After making changes to the Connection String and changing the path to the XML RSS feeds that present the most current blogs on the Home and Blogs page, the ASP.NET MVC 4 Web Role project is ready for deployment.

To publish the ASP.NET MVC 4 website to the Windows Azure Cloud Service, complete the following steps:

1. **Sign-in and download Windows Azure credentials.** To begin the publishing of the Web Role, right-click the project name and select Publish to Windows Azure. Figure 6-26 displays this menu item.

www.it-ebooks.info

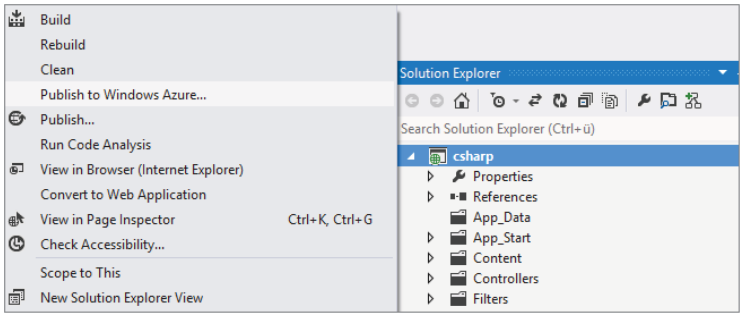


FIGURE 6-26

2. **Enter your download credentials.** When the Publish Windows Azure Application window is rendered, as shown in Figure 6-27, select the Sign In to Download Credentials link, and enter your Windows Azure credentials. When successfully authorized, you are prompted to download a *.publishsettings file.

3. **Save the publish file.** Save this file to your local computer, and, for later reference, write down where it is saved. Select the Import button (refer to Figure 6-27) and select the *.publishsettings file you just downloaded. Then select Next to move to the next window. Figure 6-28 illustrates the next window.

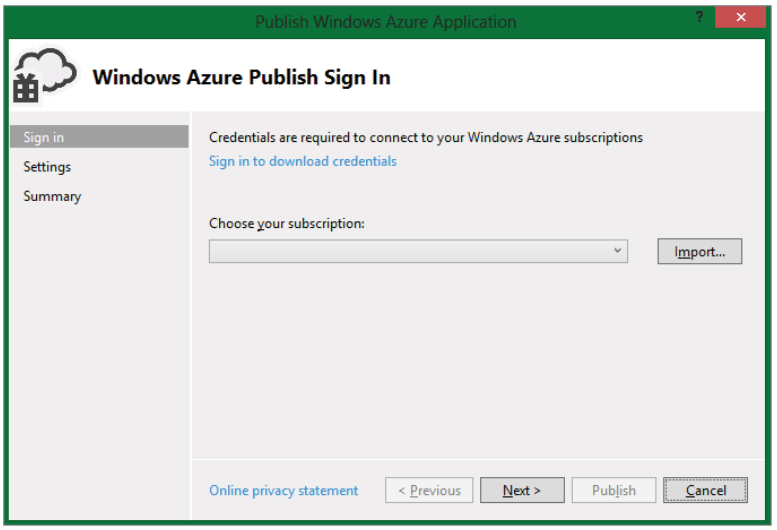


FIGURE 6-27

www.it-ebooks.info

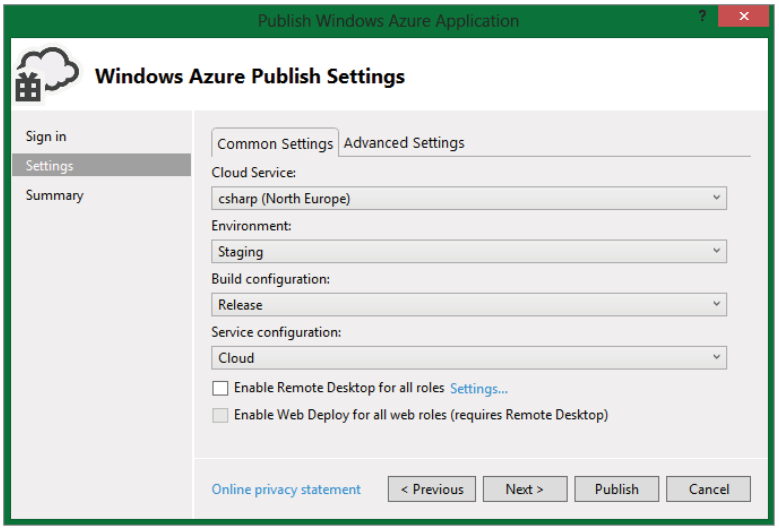


FIGURE 6-28

NOTE *With the release of the Windows Azure SDK 2.0, you can now associate your Windows Azure Subscription directly with Visual Studio. Instead of downloading and importing the publishing file, you can select the specific website to which you want to publish.*

4. **Publish the ASP.NET MVC 4 website.** Select the Cloud Service that you created earlier in the chapter, chose Staging for the Environment, and leave the remaining default values as they are.
5. **Review the summary.** Select the Next button to review the Summary.
6. **Deploy the Web Role.** Select the Publish button to deploy your Web Role to the Windows Azure Platform. Figure 6-29 is an example of the deployment status window rendered after the Publish button is selected.

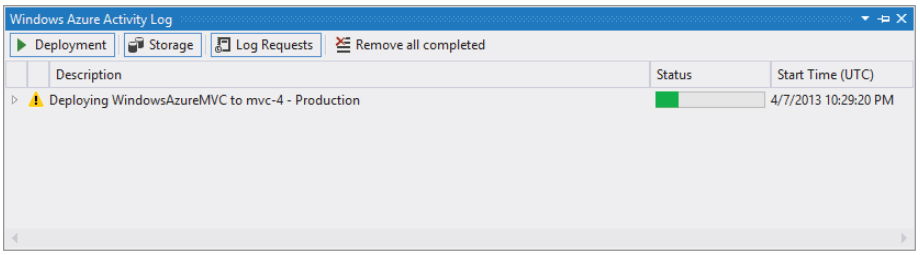


FIGURE 6-29

www.it-ebooks.info

NOTE *You may want to give the migration a few minutes to complete its installation and configuration before you begin testing if the deployment were successful.*

7. **Review the result of the published website.** When you are ready to begin testing the deployment, open the Windows Azure management console, and select the Cloud Services menu item. Then select the Cloud Service you created at the beginning of this chapter. Figure 6-30 shows this page.

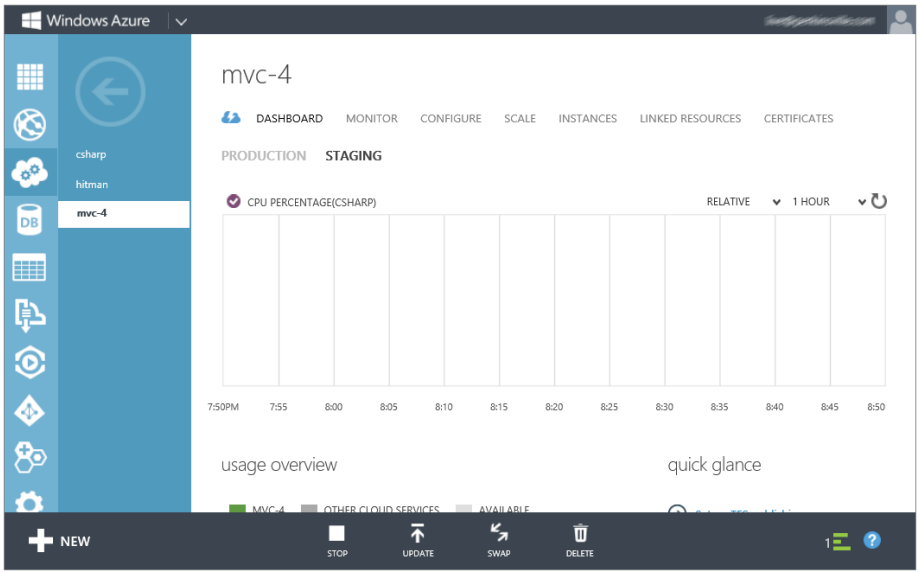


FIGURE 6-30

8 **Select the URL to test.** On the lower-right side of the under the Quick Glance section, look for the URL under SITE URL. Use this URL for testing the Staging deployment.

9. **Navigate through all the pages.** Test all the functionalities to confirm that the deployment of the Web Role was successful. When confirmed, you can move the Staging environment to Production.

10. **Swap from Staging to Production.** Select the Swap button on the lower middle of the page, and you are prompted with a message box resembling that in Figure 6-31. Select Yes and watch the status bar for deployment updates and completion notification.

www.it-ebooks.info

FIGURE 6-31

11. **Confirm the Web Role has migrated.** When complete click the Production link to confirm that the Web Role has been migrated from Staging to Production, as shown in Figure 6-32. Then search for the Production URL under the heading of SITE URL and validate the Web Role responds and everything is okay with the Cloud Service.

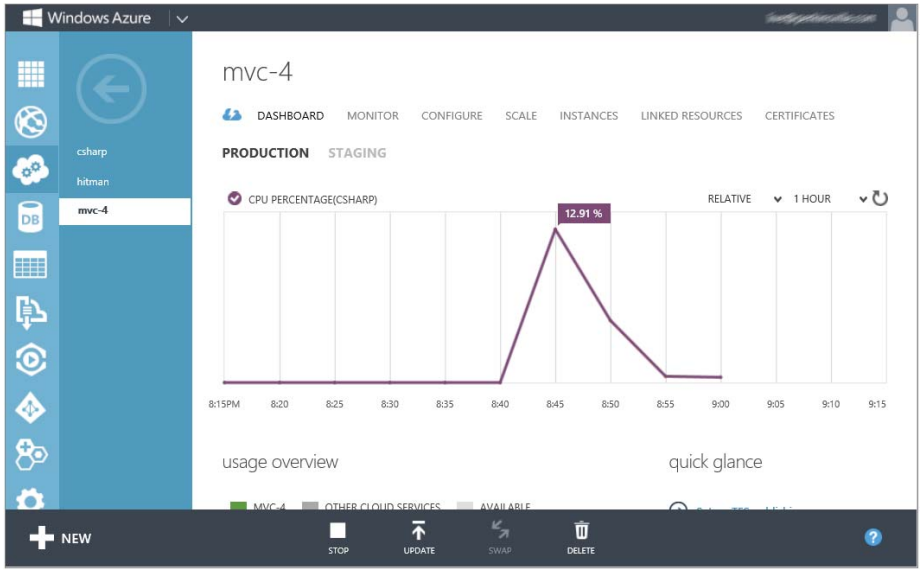


FIGURE 6-32

For this example the SITE URL is `http://mvc-4.cloudapp.net` and you can access it to compare it against your migrated website. At this point you have completed the migration of the sample ASP.NET MVC 4 Web Role to the Windows Azure platform. To perform the same from Team Foundation Server (TFS), continue to the next section.

www.it-ebooks.info

SETTING A TFS CONNECTION AND PUBLISHING THE ASP.NET MVC 4 WEBSITE

In this section you set up a link between the ASP.NET MVC 4 website created earlier in this chapter and Team Foundation Server. As you learned in the previous chapter, Team Foundation Server is not only useful for source code storage and control, but also for managing team workflow assignments, application lifecycles, and so on.

Implementing Team Foundation Server into your company is not only a change in the way you deploy your code from a local machine onto the Windows Azure platform, but also to the way you manage your entire project management and software development processes. So be sure you have read about Team Foundation Server in Chapter 5; plus, review the numerous resources found on the Internet. The information covered in this section discusses only how to set up the connection between TFS and a website hosted on Windows Azure.

To set up a link between a website hosted on Windows Azure and Team Foundation Server, follow the steps in these sections.

Accessing the Team Foundation Server

Before you can publish your ASP.NET MVC website to Windows Azure using Team Foundation Server you must have access to Team Foundation Server and a project to publish your site. To get a Team Foundation Server account, follow these steps:

1. **Create an account.** After logging into the Windows Azure management console at: `http://manage.windowsazure.com`, navigate to a website that you have already created and click the Set Up TFS publishing link on the right side of the page. The pop-up wizard appears and provides you the opportunity to create a New User account. You click the Create a TFS Account Now link to get your account.

NOTE *You can also sign up for a TFS account from within Visual Studio 2012 by selecting View ➪ Team Explorer and clicking the Sign Up for Team Foundation Service.*

---

2. **Connect Visual Studio to your TFS.** After your account is created, open Visual Studio 2012, and select Team ➪ Connect to Team Foundation Server. Click the Servers button, then click Add, and enter your URL, as shown in Figure 6-33. Click OK to open the Add/Remove Team Foundation Server window.
3. **Enter your credentials.** You are prompted to enter your Microsoft ID credentials. Provide the credentials and the server is successfully added to the Team Foundation Server list, as shown in Figure 6-34. Select the Close button, and, on the following window, select the Connect button.

www.it-ebooks.info

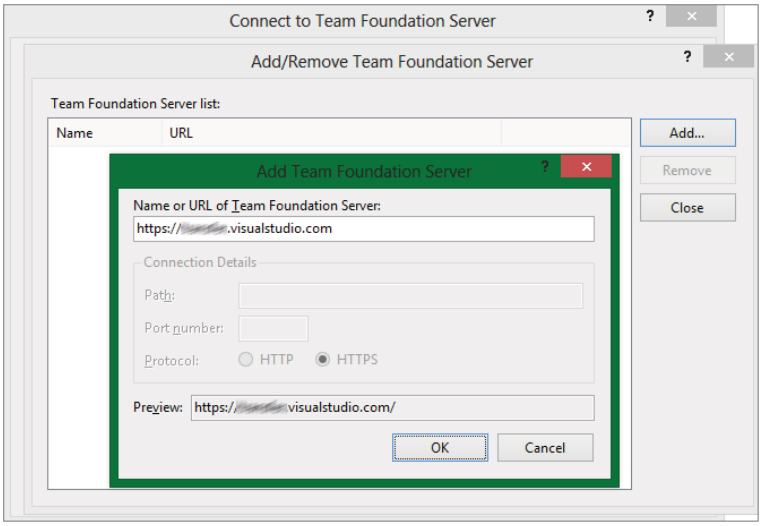


FIGURE 6-33

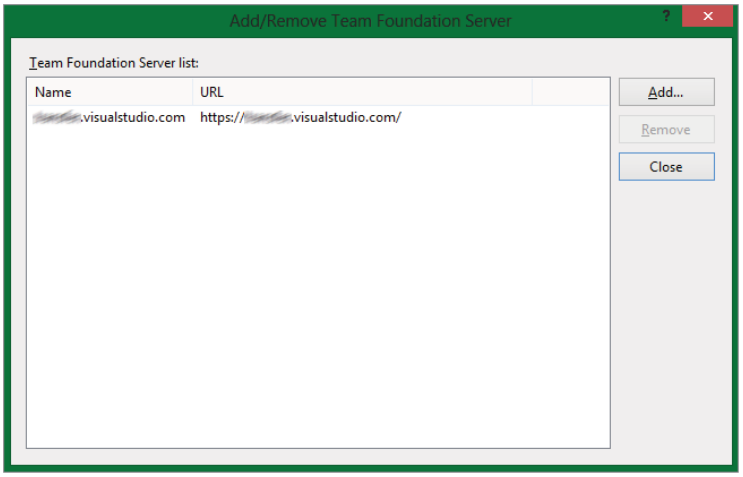


FIGURE 6-34

NOTE *When connected from Visual Studio 2012, the Team Explorer windows open on the right side of the IDE. If this is not the case, select View ➪ Team Explorer from the menu or press Ctrl+M to open it.*

4. **Create a TFS Project.** Select the Create a New Team Project link to start the Project Creation Wizard in your TFS environment. Figure 6-35 shows the first window in the wizard.

www.it-ebooks.info

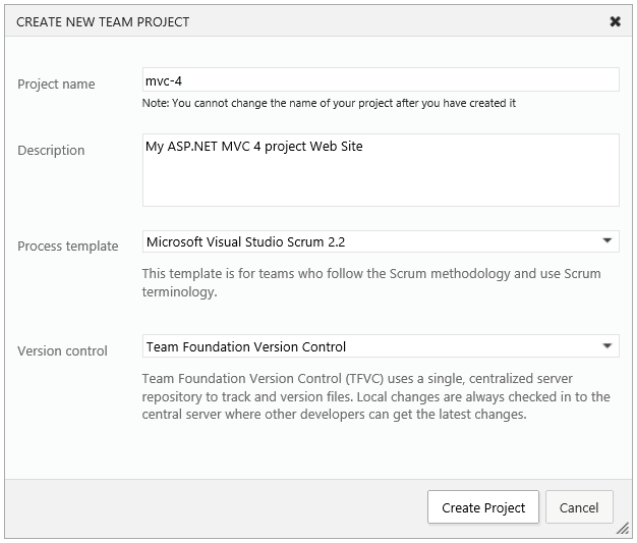

CREATE NEW TEAM PROJECT
Project name
mvc-4
Note: You cannot change the name of your project after you have created it
Description
My ASP.NET MVC 4 project Web Site
Process template
Microsoft Visual Studio Scrum 2.2
This template is for teams who follow the Scrum methodology and use Scrum terminology.
Version control
Team Foundation Version Control
Team Foundation Version Control (TFVC) uses a single, centralized server repository to track and version files. Local changes are always checked in to the central server where other developers can get the latest changes.
Create Project
Cancel

FIGURE 6-35

5. **Enter project information.** Enter the information shown in Figure 6-35, and select the Create Project button. When selected, a status window is rendered, and the creation of your TFS project is run to completion.

6. **View your project overview.** When complete click the Navigate to Project button that provides an overview of your just-created project.

NOTE *You need an Internet connection to complete these and many of the previous steps in this section. They cannot be performed offline.*

---

7. **Add the ASP.NET MVC 4 source code to the TFS project.** Return to Visual Studio 2012, and select the Connect to a Team Project link, as shown in Figure 6-36. Select the project that was just created from the Team Projects list box; select the Connect button.

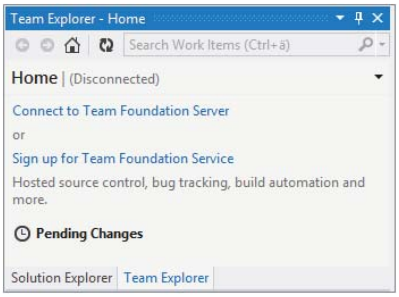


FIGURE 6-36

www.it-ebooks.info

8. **Add code to your project.** Select the Source Code Explorer link, as illustrated in Figure 6-37. This opens up a window that allows you to add code to the mvc-4 project. Right-click the mvc-4 project in the Folder pane, and select Add Items to Folder, as shown in Figure 6-38. Navigate to the location where you saved the ASP.NET MVC 4 project, and select the folder, as shown in Figure 6-39. Select the Map button on the same window, and make sure the path is the same as the location just provided. Press the Map button. Then click Next.

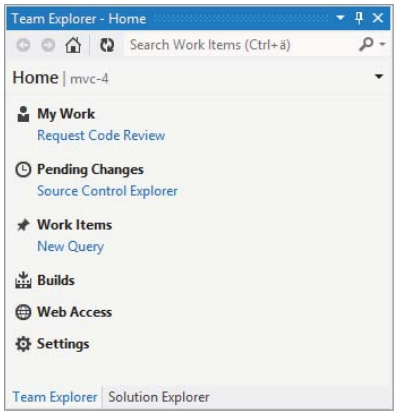


FIGURE 6-37

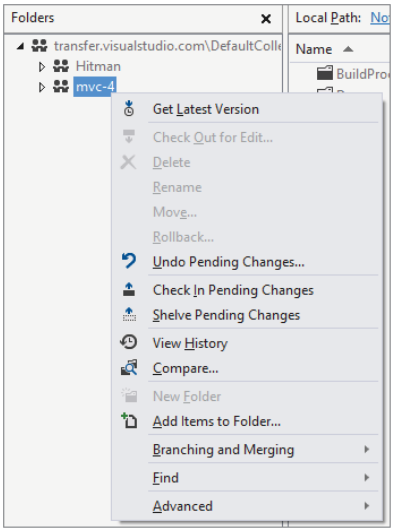


FIGURE 6-38

www.it-ebooks.info

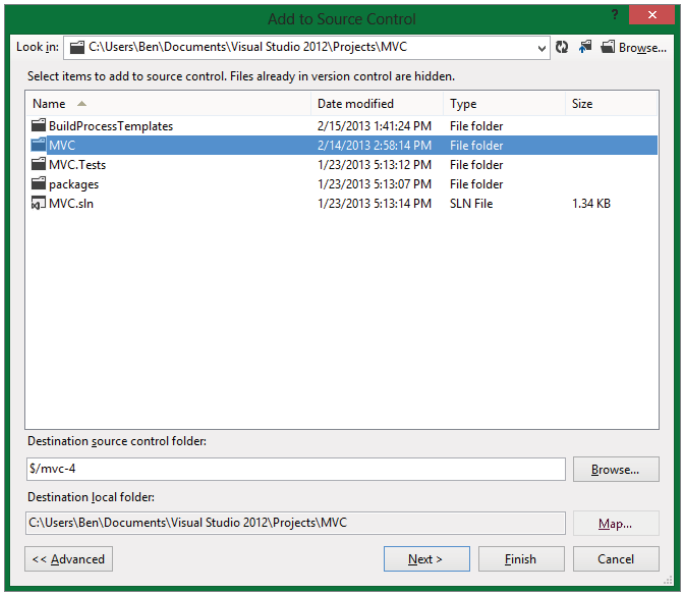


FIGURE 6-39

9. **Review the code files you're adding.** Selecting the Next button presents the files to be added to the project. Review the files and press the Finish button to add the files to the project.

You can now expand the mvc-4 project and view the files you just added. Figure 6-40 shows the files added to the project.

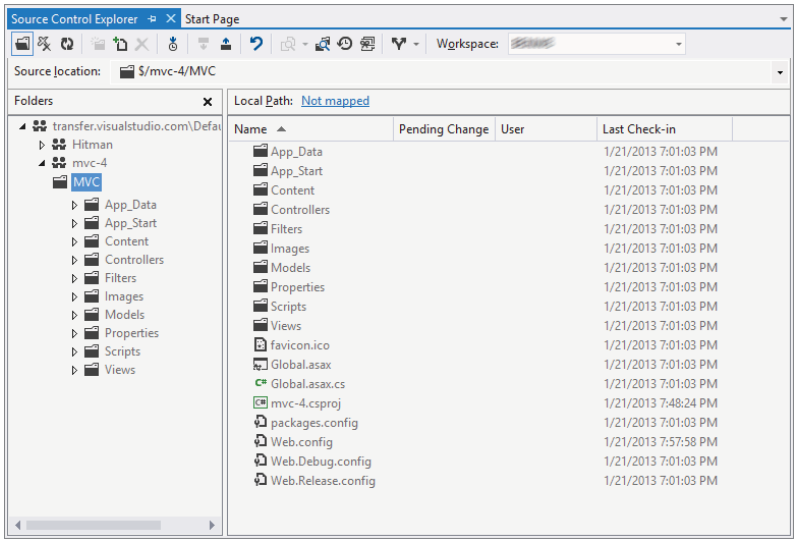


FIGURE 6-40

www.it-ebooks.info

Adding Links Between TFS and Windows Azure

Up to now, you have created a TFS account, a TFS project, and you've added the ASP.NET MVC 4 source code to the TFS project. Now you must link the TFS project and Windows Azure. Without linking these two accounts together, deployment of the code is not possible.

1. **Connect the TFS project just created to Windows Azure.** In the Windows Azure management console, click the website you want to create the link with.
2. **Authorize the connection.** Click the Set Up TFS Publishing link, as shown in Figure 6-41. When clicked, a pop-up window is rendered and walks you through the step to authorize the connection. Figure 6-42 shows the first page of this wizard. If you do not already have a TFS account, click the Create a TFS Account Now link to select an account URL and link it to your Microsoft ID and create the account.

NOTE *If you do already have a TFS account, enter the user ID into the wizard, and click the Authorize Now link ➪ Accept the Request for Permissions, if you want. (It is required to continue.)*

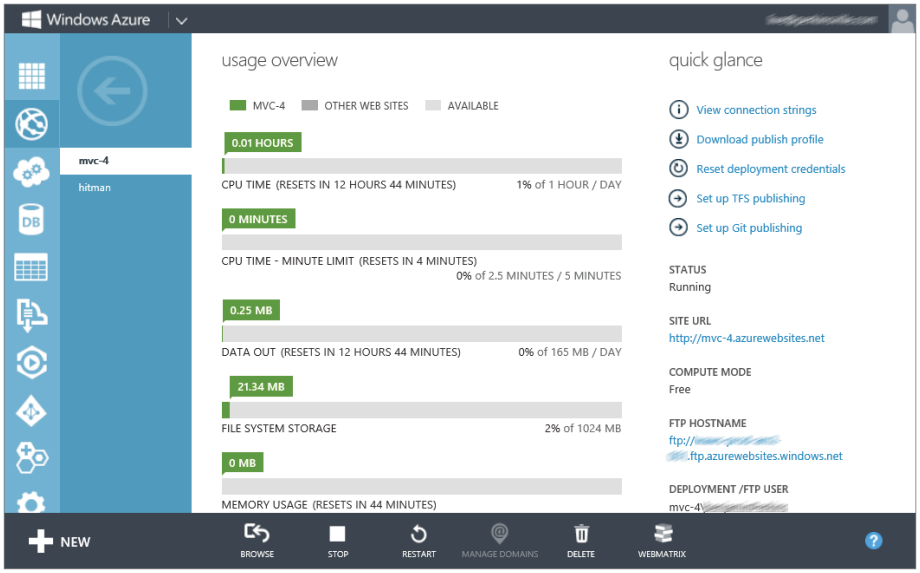


FIGURE 6-41

3. **Create the link.** As illustrated in Figure 6-43, you can select the TFS link you just created in step 2. Click the check on the lower-right side of the window to complete the link between Team Foundation Server and the Windows Azure Web Site.

www.it-ebooks.info

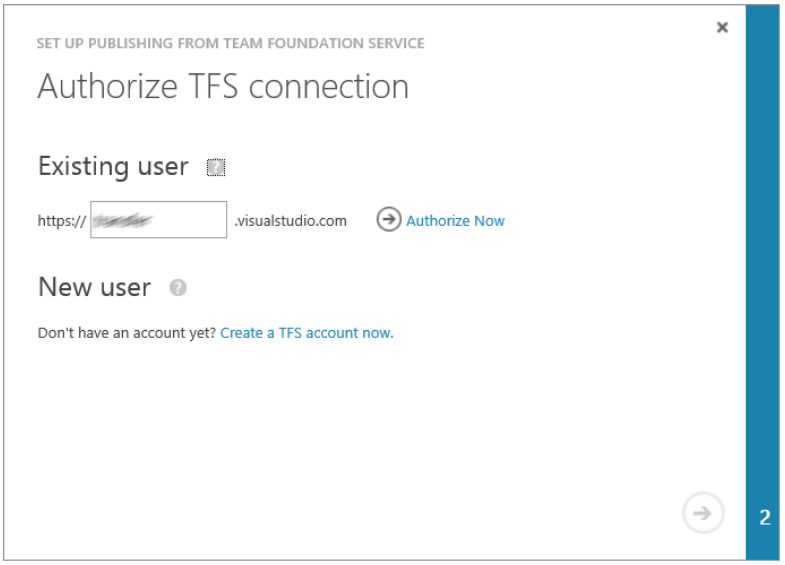


FIGURE 6-42

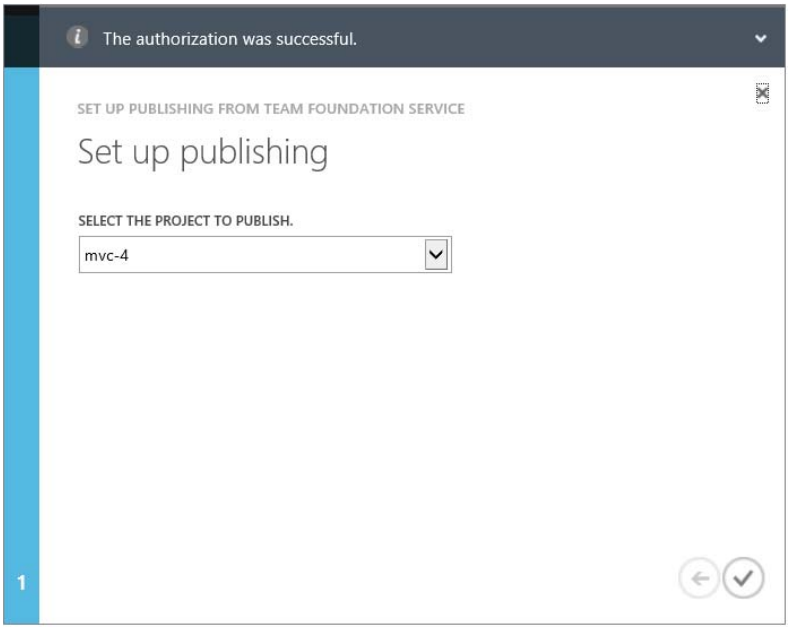


FIGURE 6-43

www.it-ebooks.info

After the link is completed, the page shown in Figure 6-44 is rendered.

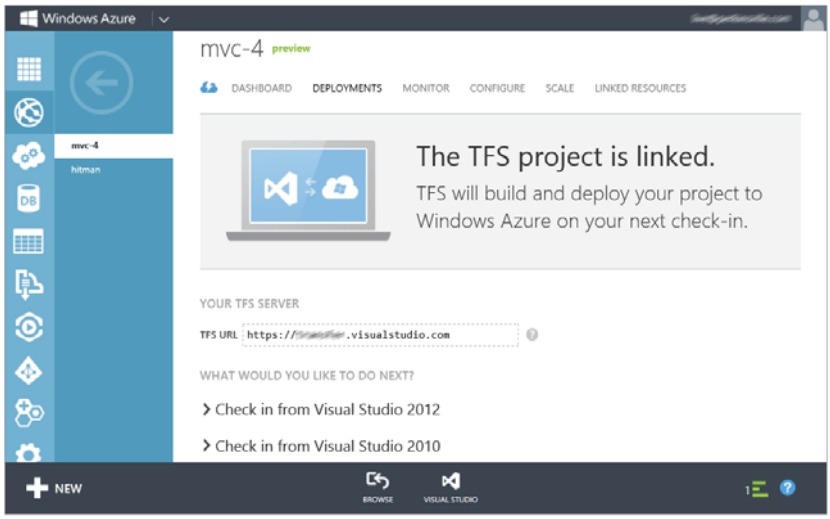


FIGURE 6-44

4. **Select a Visual Studio version.** Click the link appropriate to your version of Visual Studio. This example has been using Visual Studio 2012; follow the instructions shown. You can return to these instructions at any time by selecting the Deployments link found on the Web Site details page on the Windows Azure management console.

5. **Check in your project to publish it.** Open Visual Studio and connect to Team Foundation Server. Select the mvc-4 (or your project) to open it, and then, in the Team Explorer window, click the Pending Changes item, as shown in Figure 6-45. Enter a Comment and press the Check In button and Yes. Once the project is built without compile errors, the project is deployed to Azure.

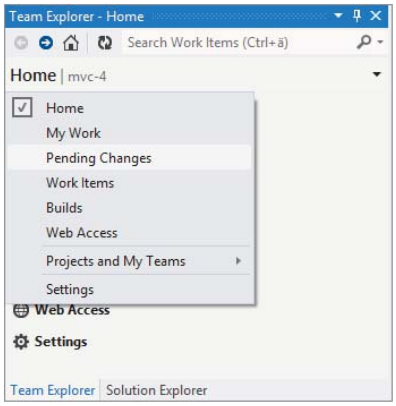


FIGURE 6-45

www.it-ebooks.info

6. **Wait for the code to be checked in and published to the mvc-4 Windows Azure Web Site.** This can take a while, so please be patient. When check in is complete, confirm your SITE URL has been published by entering it into a browser and pressing the Enter key.

NOTE *One thing to remember about this process is that each time you make changes and check them in, the content is published to the linked website. There are configurations that can change the frequency of publishing, as mentioned in the previous chapter.*

In the next section, you learn how to connect your Windows Azure Web Site to Git.

CONNECTING A WINDOWS AZURE WEB SITE TO A GITHUB CODE REPOSITORY

There are alternative source code integration platforms to Team Foundations Server. One such platform is GitHub located at `https://github.com/`. Like TFS, GitHub is a place that you can use to store your source code and manage its versions, state, and migration. GitHub is used by many open source projects and if you make your source code public, the service is free. TFS does currently provide up to five accounts for free, however, if more developers require access, then there is a charge. To set up GitHub to publish to your Windows Azure Web Site, perform the following:

NOTE *To complete this section, you need an FTP user ID and password that is created in the section titled "Setting Up FTP Capability." You might consider completing that section before starting this one.*

---

1. **Disconnect from TFS.** You can have only a single source code repository associated with your Windows Azure Web Site, so if you are still linked to TFS from the previous exercise, you need to disconnect from it before you can enable Git. To disconnect from TFS, access the Dashboard of the mvc-4 website you previously used and select the Disconnect from TFS link found in the Quick Glance section. After the connection is broken, both options for connecting to TFS and Git return.
2. **Select the Set up Git publishing link.** The following page, as shown in Figure 6-46, is rendered after the Hub is created. Write down the GIT URL for use in Step 6.
3. **Set up the GitHub project.** Go to `http://github.com` and create an account if you do not already have one. Log in and create a new repository, as illustrated with Figure 6-47.

www.it-ebooks.info

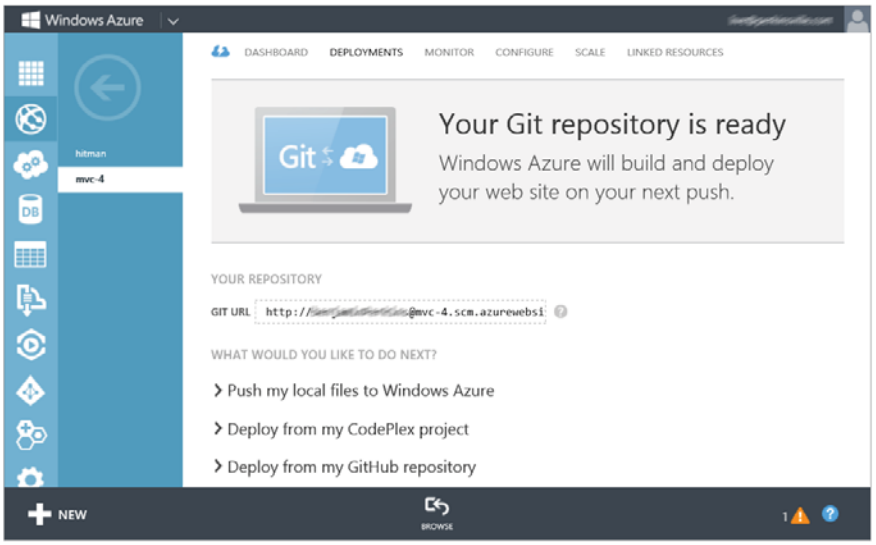


FIGURE 6-46

FIGURE 6-47

4. **Create the repository.** On the create repository page that appears, enter data similar to that shown in Figure 6-48 and then click the Create Repository button.

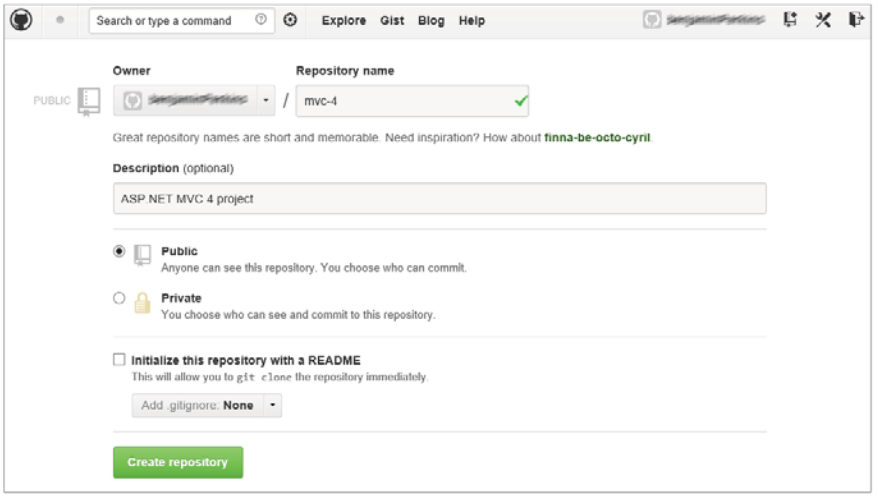


FIGURE 6-48

www.it-ebooks.info

5. **Download and install GitHub.** Click the Clone in Windows button, as shown in Figure 6-49. This takes you to a website where you can download GitHub. Once you download and install GitHub, you must enter your credentials in the window shown in Figure 6-50. Then, continue to the next window and enter in your full name and e-mail address, as shown in Figure 6-51. You'll see the GitHub repository you just created.

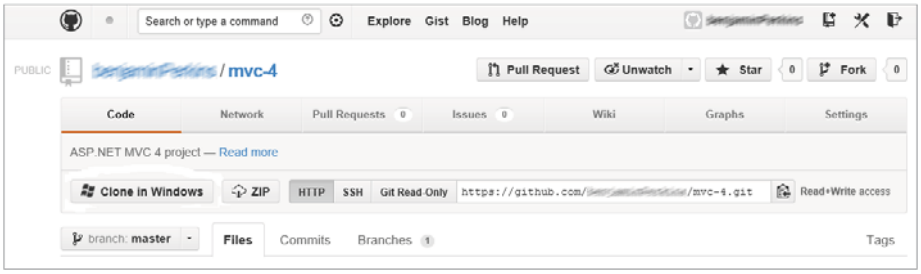


FIGURE 6-49

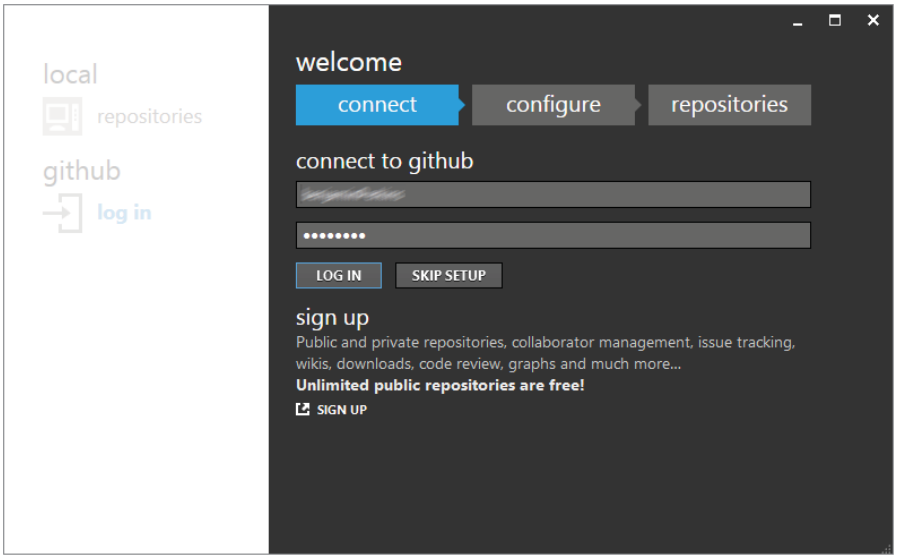


FIGURE 6-50

6. **Connect the GitHub with the Windows Azure Web Site.** Right-click the mvc-4 repository and select Open a Shell Here, as shown in Figure 6-52, from the pop-up menu. In the shell, enter the command shown in Figure 6-53 to connect the GitHub with the Windows Azure Web Site. Notice that the HTTP address is the same as shown in Figure 6-46.

www.it-ebooks.info

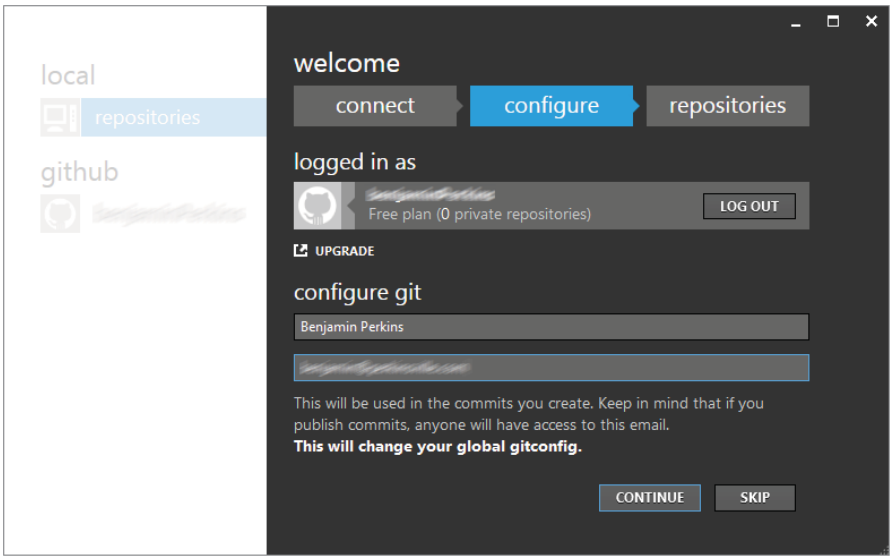


FIGURE 6-51

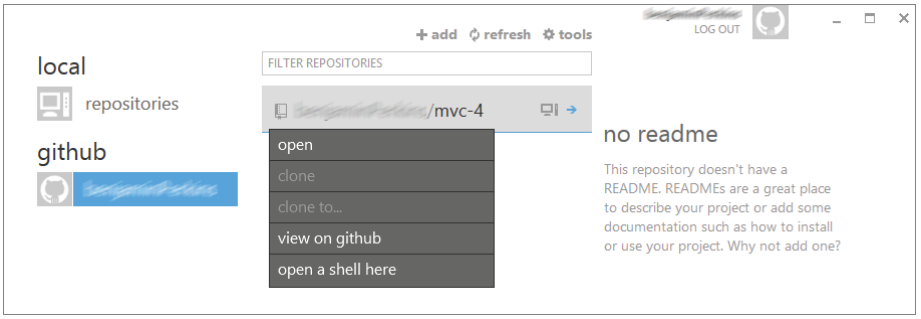


FIGURE 6-52

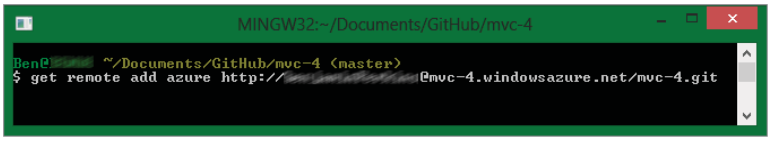


FIGURE 6-53

7. **Publish to Windows Azure.** Enter this command: `git push azure master`. This prompts you for your FTP user ID and password, created in the next section, and the site will be published to Windows Azure.

www.it-ebooks.info

PUBLISHING AN ASP.NET MVC 4 WEBSITE USING FTP

A very common approach for individuals and some companies to publish code from a local machine to a website is by using an FTP application. One of the most popular open source FTP software programs today is FileZilla. It can be downloaded from this Internet address: `http://sourceforge.net/projects/filezilla/`.

FileZilla enables the user to connect to an FTP address and add, remove, rename, and copy files in a similar way that is supported in Windows Explorer. Figure 6-54 shows a snapshot of FileZilla.

Using an FTP application such as FileZilla is actually one of the easiest ways to publish your website to Windows Azure. It works just like it would at any other web-hosting company.

The following sections show you how to set up the FTP application, and then how to publish your project.

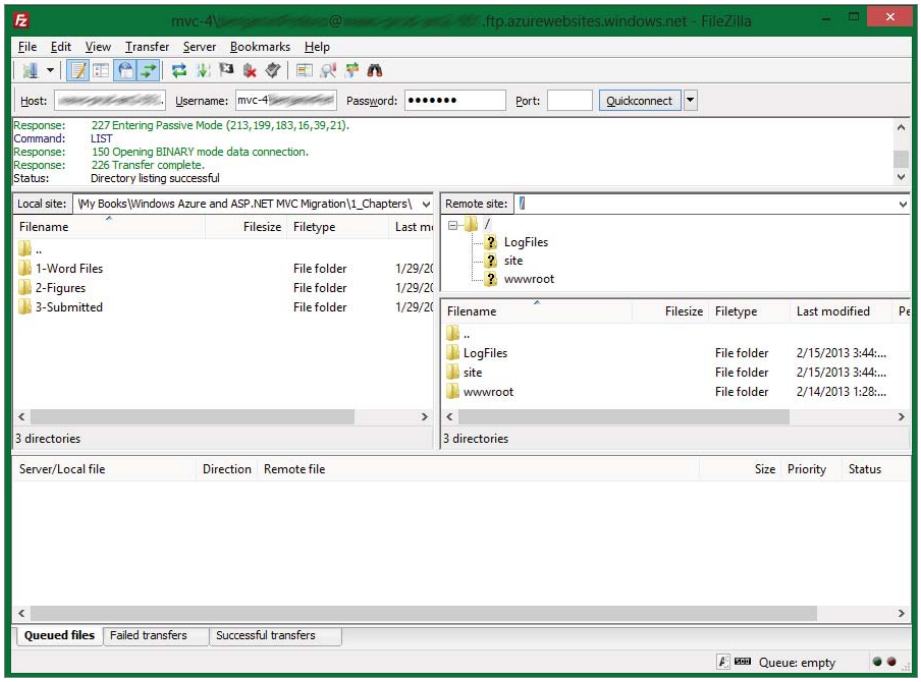

mvc-4\ @ .ftp.azurewebsites.windows.net - FileZilla
File Edit View Transfer Server Bookmarks Help
Host: Username: mvc-4 Password: ••••••• Port: Quickconnect
Response: 227 Entering Passive Mode (213,199,183,16,39,21).
Command: LIST
Response: 150 Opening BINARY mode data connection.
Response: 226 Transfer complete.
Status: Directory listing successful
Local site: \My Books\Windows Azure and ASP.NET MVC Migration\1_Chapters\
Filename Filesize Filetype Last m
1-Word Files File folder 1/29/2(
2-Figures File folder 1/29/2(
3-Submitted File folder 1/29/2(
3 directories
Remote site: /
LogFiles
site
wwwroot
Filename Filesize Filetype Last modified
LogFiles File folder 2/15/2013 3:44:...
site File folder 2/15/2013 3:44:...
wwwroot File folder 2/14/2013 1:28:...
3 directories
Server/Local file Direction Remote file Size Priority Status
Queued files Failed transfers Successful transfers
Queue: empty

FIGURE 6-54

www.it-ebooks.info

Setting Up FTP Capability

Using the FTP protocol is a good choice for individuals or companies with smaller websites who do not want or need to manage all the complexities of GitHub or TFS. To set up FTP capabilities on the Windows Azure platform, complete the following actions:

1. **Create your deployment credentials.** To create or modify your deployment credentials, select the Create/Reset Deployment Credentials link found under the quick glance section. A window, shown in Figure 6-55, appears where you enter the username for your FTP account and then your password, and then click the check mark on the lower-right.

NOTE *It is not possible to use your Microsoft ID to perform an FTP deployment.*

---

2. **Retrieve the FTP HOSTNAME.** Navigate down the right side of the website Dashboard until you find the heading FTP HOSTNAME. Under this heading is the FTP address you can use to publish to your Windows Azure Web Site.

You now have all the information required for making an FTP connection to your website hosted on Windows Azure. The required information is the FTP username, the FTP password, and FTP host name.

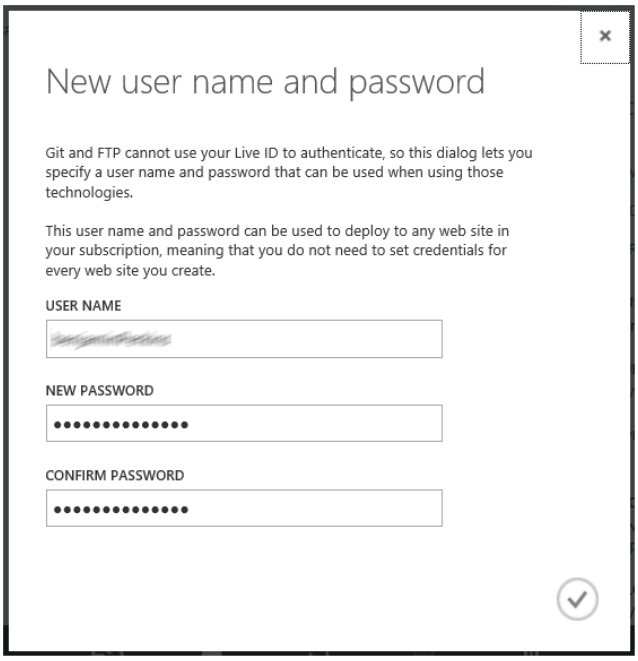

New user name and password
Git and FTP cannot use your Live ID to authenticate, so this dialog lets you specify a user name and password that can be used when using those technologies.
This user name and password can be used to deploy to any web site in your subscription, meaning that you do not need to set credentials for every web site you create.
USER NAME
NEW PASSWORD
CONFIRM PASSWORD

FIGURE 6-55

www.it-ebooks.info

Publishing the Project

To publish the ASP.NET MVC 4 project to the mcv-4 Windows Azure Web Site perform the following steps:

1. Open FileZilla using the credentials and FTP HOSTNAME created previously.
2. In the Local Site window in FileZilla, navigate to the MVC project. As an example, on my machine the directory is `C:\Users\Ben\Documents\Visual Studio 2012\Projects\MVC\MVC\`.
3. Select all the files in the directory in step 2 using Ctrl+A.
5. In the Remote Site window, navigate to `/site/wwwroot/`.
6. Select the Upload button and the files are transferred to that directory.
7. When the upload is complete, test your website by navigating through the pages making sure all the links and functionality work as expected.

WARNING *Don't forget to comment out the code that created the database structure from the Global.asax.cs file after the website has been run once and then republish.*

SUMMARY

In this chapter, you saw how to create a Windows Azure account, the requirements for creating a Cloud Service and website, how to create a SQL database and multiple methods for publishing your website and Web Role to the Windows Azure platform. You also learned the requirements for converting an ASP.NET MVC 4 website project to a Windows Azure Web Role using the Windows Azure SDK for .NET template.

This chapter showed that although publishing your code using a source code repository, such as Team Foundation Server or GitHub, requires only a few steps, the process and organization required for getting the code into position to deploy can take significant effort. For example, using Scrum teams and workflow processes can impact the frequency of deployments as well as the development and integration testing requirements.

Now that you have created the ASP.NET MVC 4 project, fine-tuned it for optimal performance, and deployed it to the Windows Azure environment, you should know how to monitor, maintain, and troubleshoot issues that may happen after your website is live. The next chapter covers ideas and techniques you need to perform many of these tasks.

www.it-ebooks.info

PART IV
Monitoring and Troubleshooting

- **CHAPTER 7:** Maintaining an ASP.NET MVC 4 Deployment on Windows Azure
- **CHAPTER 8:** Monitoring and Supporting an ASP.NET MVC 4 Project on Windows Azure

www.it-ebooks.info

7 Maintaining an ASP.NET MVC 4 Deployment on Windows Azure

CONCEPTS

IN THIS CHAPTER

- How to monitor a Windows Azure Web Site
- How to monitor a Windows Azure Cloud Service
- How to use Windows Azure tools

The life of an application begins after development, optimization, and deployment. When customers and users access the system's features and services, they'll notice its behavior and personality. In some cases, the reasons for a problem or behavior are obvious and you can easily correct or modify them. Other complications are more subtle and harder to understand, track, trace, and log. To track, trace and log, it was once a common practice in ASP-based websites to use print statements for debugging, or to strategically place `Response.Write()` commands in the file to write algorithm results and to confirm run time values.

Debugging and troubleshooting capabilities have improved greatly with the introduction of such technologies as System.Diagnostics, Tracing, ETW, Logman, Failed Request Tracing, and so on, which are built into the Windows Server environment and require no additional installations. This is important because in many environments security, release, or change management processes prevent you from installing tools and monitors or make it extremely time-consuming to deploy them. When production problems occur, it is important to move fast, so having these features on the platform by default moves debugging and diagnosis along quicker.

If you have completed all the chapters in the book up to now, you have deployed an ASP.NET MVC 4 project onto a Windows Azure Web Site and Cloud Service. If you have not completed all the chapter exercises but still have a website or Web Role on Windows Azure, you might still be asking a valid question: How can you monitor, log, and troubleshoot problems that happen within the application? Troubleshooting and logging capabilities differ between a website and a Web Role. This section discusses these differences and provides details about the troubleshooting features of both.

www.it-ebooks.info

MONITORING A WINDOWS AZURE WEB SITE

Windows Azure Web Site has numerous monitoring features that do not require any further modifications or additions to code. You can find these on either the Dashboard or the Azure Web Site's Console. The reasons for monitoring your system are numerous and include performance fine-tuning and usage forecasting.

A website administrator should always watch for opportunities to improve performance; however, without baseline performance measures, if you implement any change you would not have a performance metric with which to compare against. Therefore, capturing, storing and analyzing baseline metrics is an important part of monitoring.

Monitoring is also useful for forecasting usage and scaling requirements. If a website administrator sees the CPU or memory utilization of the system trending toward 100 percent utilization, he or she must increase the hardware capacity. CPU and memory utilization are both parameters that the administrator can monitor and store via the Windows Azure Dashboard.

Monitoring with the Dashboard

This Dashboard provides an overview of some basic metrics. For example, CPU Time, Data In and Out, and HTTP Server Errors are shown, by default, via a graph with the associated value. If you access the Windows Azure management console by clicking Web Sites and then the Dashboard you want to monitor, you'll see a page similar to that shown in Figure 7-1.

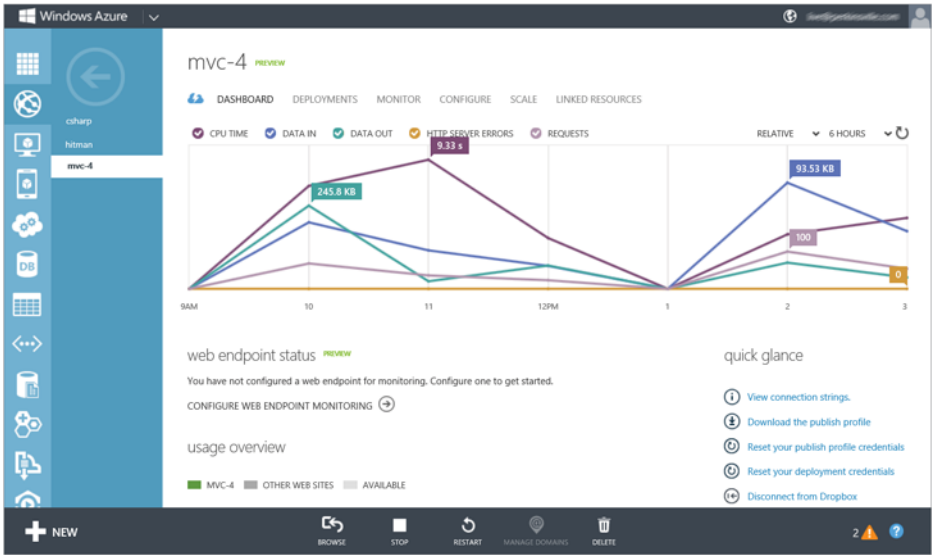


FIGURE 7-1

www.it-ebooks.info

Basic metrics include:

- **Relative versus absolute:** You can change the graph properties to show relative versus absolute values. When the Absolute option is selected, the Y-axis values display, and the graph is modified so that the values of the metrics drive the graph instead of the relative relationship between the metrics value. See Figure 7-2 as an example and compare it to Figure 7-1.

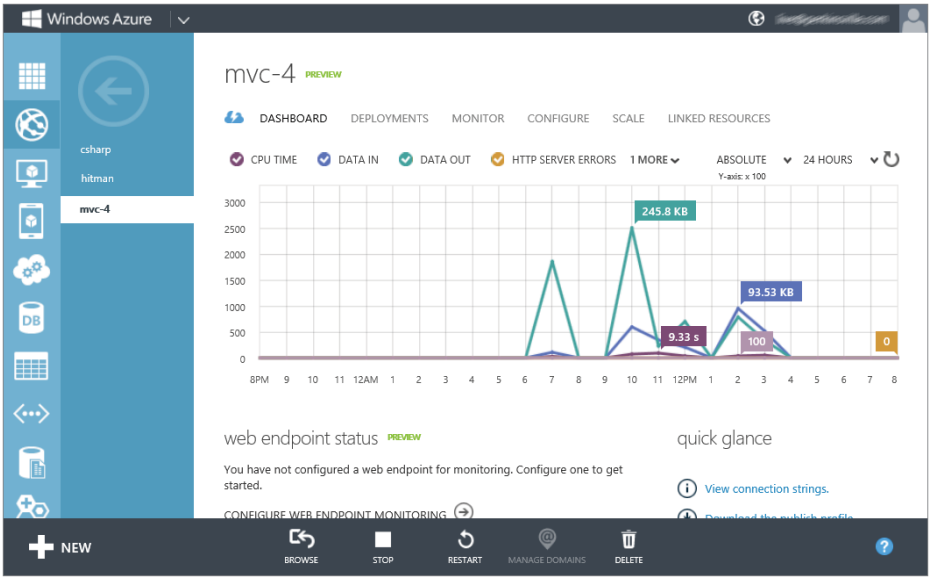


FIGURE 7-2

As you can see in Figure 7-2, the values are graphed by the measured value of the metric instead of the relative value.

- **Changing the Timeframe:** You can also change the time frame of the graph from 6 hours, 24 hours, and 7 days by selecting the desired value from the drop-down located on the top-right of the graph shown previously in Figure 7-2. This provides the administrator with a quick view of the overall current and historical usage and health of the website.

Monitoring with the Website's Management Console

Other than the Dashboard, you have another graph in the Monitor section of the Windows Azure Web Site's management console. This graph provides additional metrics that you can add by clicking the Add Metrics link at the bottom of the management console page. Selecting this link opens a pop-up menu, as shown in Figure 7-3.

www.it-ebooks.info

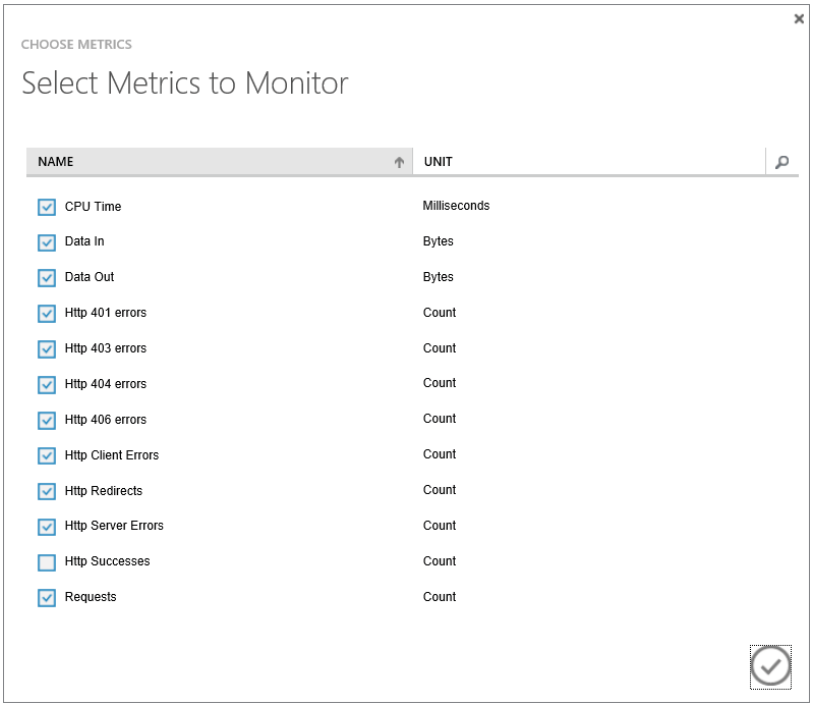


FIGURE 7-3

These metrics give you some quick but deeper insight into your website without you needing to download and analyze any logs or traces. Some of the traceable metrics include authentication errors, attempted access to restricted files, requests for files or services that were not found, and server errors. These metrics and graphs are useful; however, if you notice something that doesn't seem right, like a lot of authentication or server error counts, deeper analysis of the issue is required.

When you identify that there may be a problem on the website and want to dig a little deeper into the problems, you have the following options:

- **Application Logging (File System).** This is similar to the Application Event logging that is commonly accessed through the Event Viewer. The logging level is configurable so that only error, warning, information or verbose errors are logged. This log provides time, date, an Event ID, a stack dump, and often the specific component that caused the event to be logged. This information is useful when you're debugging any issue.
- **Application Logging (Storage).** Similar to the Application Logging (File System) feature, but instead the logs are written to a storage account so that the data can be accessed remotely using a database connection string. You can apply filters to the data as well as create pivot tables to make the website performance analysis simpler to track and monitor.

www.it-ebooks.info

- **Site Diagnostics.** This section contains three logging capabilities that you can enable or disable from either the Windows Azure Management console or from within Visual Studio:
  - **Web Server Logging log:** Similar to the IIS log that saves requests in the W3C extended file format. This log file contains dates, times, IP addresses, port numbers, HTTP verbs, file names, files sizes, and HTTP statuses to name a few — all of which are useful pieces of information.
  - **Detailed Error Message logs:** These are .HTM files similar to that shown in Figure 7-4. The file contains the name of the file requested, the likely cause of the error, and some possible solutions.

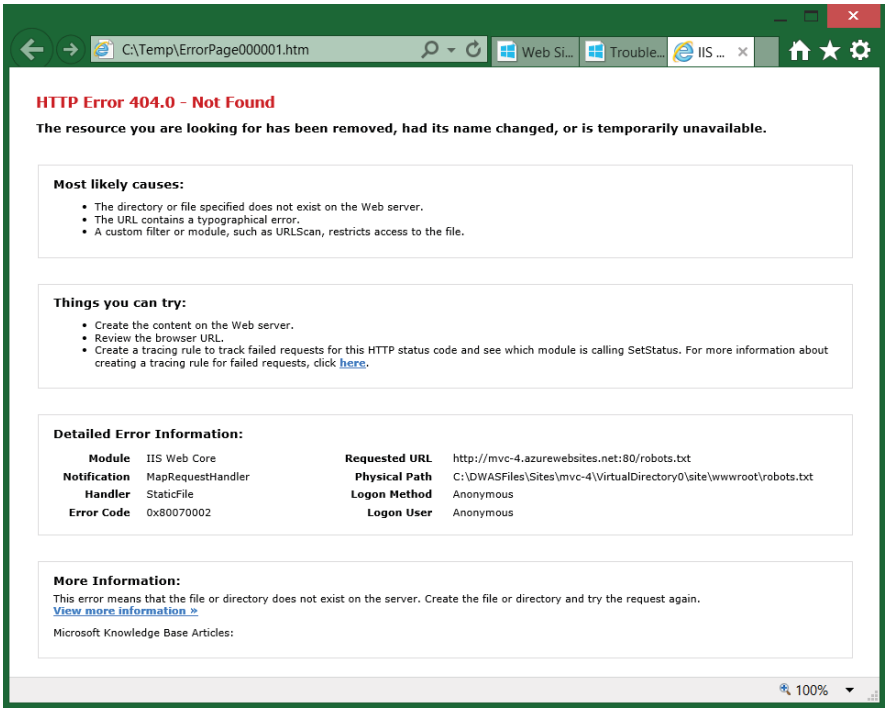

C:\Temp\ErrorPage000001.htm
Web Si...
Trouble...
IIS ...
HTTP Error 404.0 - Not Found
The resource you are looking for has been removed, had its name changed, or is temporarily unavailable.
Most likely causes:
The directory or file specified does not exist on the Web server.
The URL contains a typographical error.
A custom filter or module, such as URLScan, restricts access to the file.
Things you can try:
Create the content on the Web server.
Review the browser URL.
Create a tracing rule to track failed requests for this HTTP status code and see which module is calling SetStatus. For more information about creating a tracing rule for failed requests, click here.
Detailed Error Information:
Module IIS Web Core
Notification MapRequestHandler
Handler StaticFile
Error Code 0x80070002
Requested URL http://mvc-4.azurewebsites.net:80/robots.txt
Physical Path C:\DWASFiles\Sites\mvc-4\VirtualDirectory0\site\wwwroot\robots.txt
Logon Method Anonymous
Logon User Anonymous
More Information:
This error means that the file or directory does not exist on the server. Create the file or directory and try the request again.
View more information »
Microsoft Knowledge Base Articles:
100%

FIGURE 7-4

- **Failed Request Tracing:** Introduced in IIS 7, this feature provides a detailed log of a request as it flows through the integrated pipeline. With it, you can see when modules load in IIS, when authentication happens, when the page loads, when the page is complete, and everything in between. This log is not only helpful in identifying a specific error, but also in determining when the website is performing slowly. You can use timings logs at each stage of the process to find the exact point where the time expended is the greatest.

www.it-ebooks.info

- **Endpoint feature:** This feature checks the responsiveness and availability of your website from numerous locations around the world. Endpoints are available with Windows Azure Web Site, but only when you're in Reserved mode, where all your websites run on a dedicated virtual machine. This feature is available in all Windows Azure Cloud Services, and more details are provided about this feature in the next section.

MONITORING A WINDOWS AZURE CLOUD SERVICE

The numerous differences between a Windows Azure Web Site and a Cloud Service have been touched on throughout this book. For example, a website is likely co-hosted on a virtual machine with other websites (a multitenant configuration), while a Cloud Service is hosted on a dedicated virtual machine. Also, a website has only a single production instance, whereas a Cloud Service offers both a production and staging environment. It is possible to create multiple websites and use one for production and another for staging, however there is no built-in capabilities for the promotion of the staging environment to production as exists when using a Cloud Service. The same is true for monitoring capabilities; some features are only available through a Cloud Service.

With the Cloud Service hosted on a virtual machine, the most significant difference and benefit is creating a Remote Desktop Connection. This is useful because you have access to the same tools and features as you would if you'd connected to a virtual machine or an actual physical server. These tools include the Task Manager, the Event Viewer, and any of the many additional features available via the Server Manager Management console.

Using the Task Manager and Event Viewer

The Task Manager lets you check the percentage of CPU used per process and the amount of memory the process is consuming. It also supports the creation of a memory dump — useful for troubleshooting hanging or crashing IIS worker processes. The Event Viewer provides access to System and Application logs. These logs can help find a cause of errors encountered during an HTTP request. Each logged event has an associated ID and a source, which you can use to identify a root cause and a possible fix.

Using IIS and PowerShell

Features such as IIS and PowerShell are available when you connect to the Cloud Service, too. In the IIS management console, you can make changes to IIS logging to capture more data or install the Advanced Logging module to increase the level and value of IIS logging. You can also enable and configure Failed Request Tracing logs. These logs provide a deeper insight into the state of IIS and the HTTP request state along the entire request pipeline from start to finish. You can make any modification to IIS on a standalone server or on a virtual machine via the remote connection.

www.it-ebooks.info

NOTE *If you have scaled your worker instances and you make a change to one of the IIS instances, those changes are not replicated to the other instances automatically. Therefore, consider getting a single instance the way you need it before increasing the instances.*

PowerShell also provides features for modifying the configuration of the website hosted on IIS. Any modification you can perform from the IIS Management console can also be performed via PowerShell. For example, the PowerShell command shown in Listing 7-1 can enable IIS Logging on a website named Default Web Site.

LISTING 7-1: PowerShell Example, Enabling IIS Logging

```
PS C:\>Import-Module WebAdministration
PS C:\>Set-ItemProperty "IIS:\Sites\Default Web Site"
                        -Name logfile.enabled
                        -value $True
```

PowerShell also has a feature for checking the availability or status of the website. By simply navigating to the IIS:\Sites directory using common command-line commands such as `cd Sites` and the `dir`, the list of websites and their status are rendered.

Using the Cloud Service Management Console

The Windows Azure Cloud Service management console is similar to the Web Site management console. As shown in Figure 7-5, there are only a number of differences.

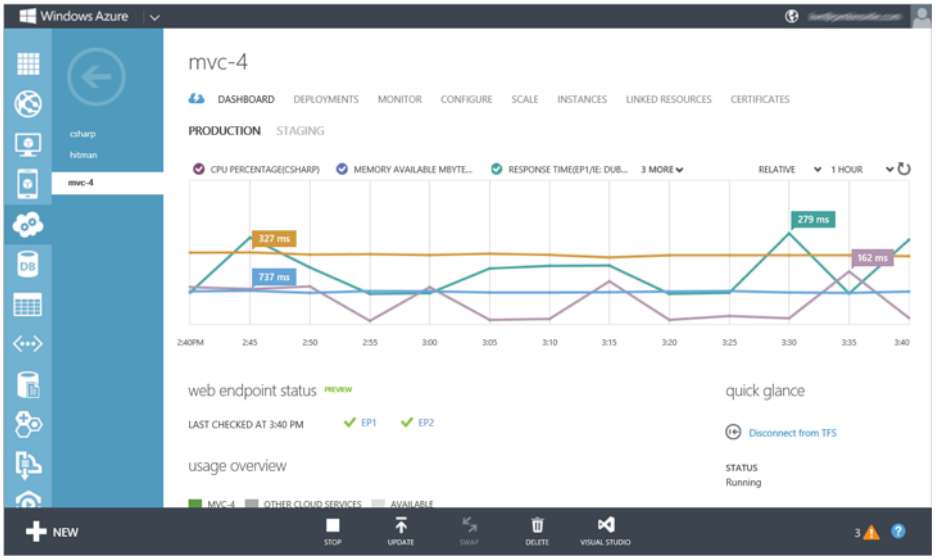


FIGURE 7-5

www.it-ebooks.info

Production and Dashboard Links

From a monitoring perspective, you might notice that the Dashboard has additional links for Production and Staging.

Timeframe

The various timeframes for setting the graph are 1 hour, 24 hours, and 7 days, whereas for a website, the smallest timeframe is 6 hours. Clicking the Monitor link provides the graph shown in Figure 7-6, which contains response times that are generated from endpoints.

FIGURE 7-6

Endpoints

As mentioned previously, both Windows Azure Web Sites and Cloud Services support endpoints; however, the website must be in Reserved Mode. For a Cloud Service, the feature is available by default. To enable an endpoint, select the Configure link, and scroll down until you see endpoints in the Monitoring group, as illustrated in Figure 7-7.

Endpoint Locations

The endpoint feature supports the input of up to three locations per two endpoints, which means you can track the Web Role performance and responsiveness from six different locations around the world. When you add this feature, select Save, and moments later the requests are made to the URLs in the endpoint, and the graph shows the results of the requests.

www.it-ebooks.info

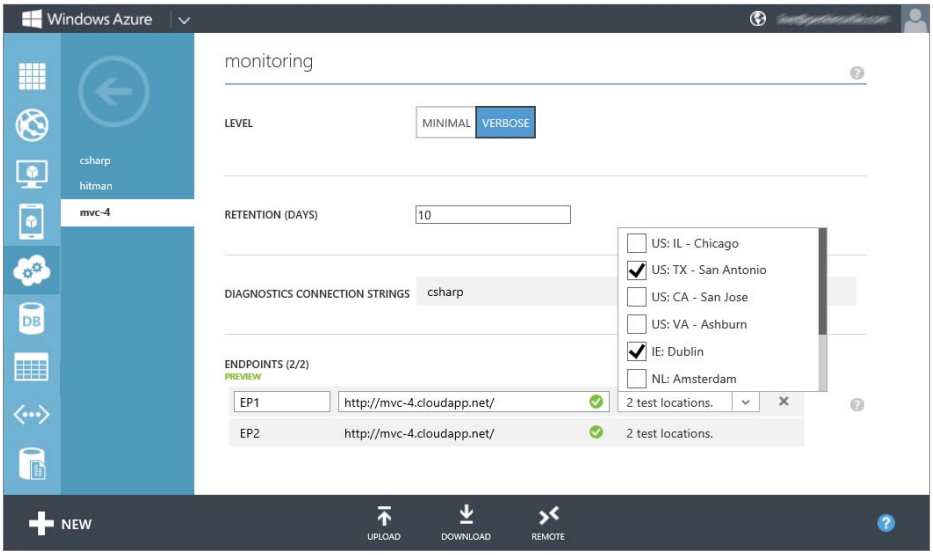


FIGURE 7-7

NOTE *Note that this feature is available in Preview mode; therefore, the limits may change at some point in the future.*

Verbose Monitoring

With this option, you can access verbose monitoring from outside of the management portal. When the monitoring level is changed from Minimal to Verbose (refer to Figure 7-7) six tables are created for the Web Role. Two tables are created for each of the following timeframes:

- 5 Minutes
- 1 Hour
- 12 Hours

To view the contents of these tables from Visual Studio 2012, follow these steps:

1. Expand the Server Explorer, and click Windows Azure Storage, as shown in Figure 7-8.
2. Enter the credentials found in the Diagnostics Connection Strings (refer to Figure 7-7). An example of the values follows:
   - `DefaultEndpointsProtocol=https`
   - `AccountName=hitman`
   - `AccountKey=ahAyTA****1hQyEEgLVi0ewk*****SVZEDhVJD+e2kK0CQaKrXRXckk uYo8S4+4/74******`

www.it-ebooks.info

NOTE *The Windows Azure SDK for Visual Studio 2012 must be installed for the Windows Azure Storage option to be available. For more information, see Chapter 6.*

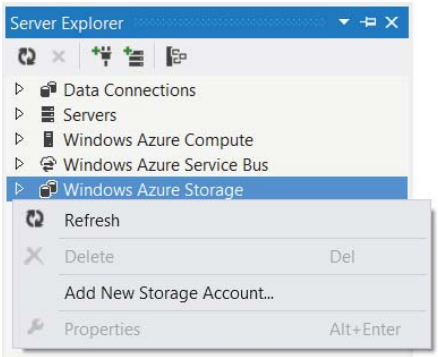


FIGURE 7-8

3. After the connection is made, you can expand the Tables listing by clicking the Account Name and then ➪ Tables. The tables containing the verbose data are rendered, as shown in Figure 7-9.

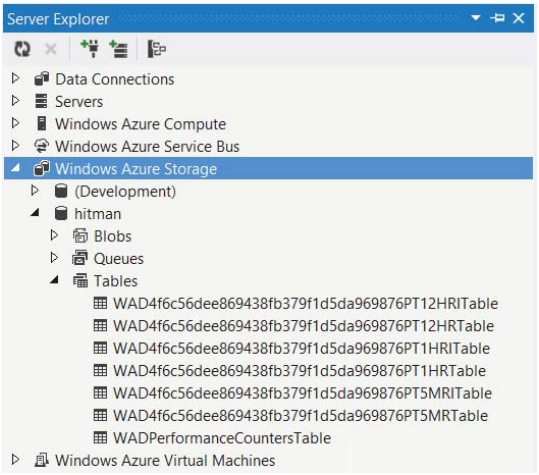


FIGURE 7-9

The format of the table is a GUID-like number and the aggregation interval (5M, 1H, or 12H) and role level (R) or role-level instances (RI). Example data stored in the 1HR table is shown in Figure 7-10.

www.it-ebooks.info

PartitionKey	RowKey	Timestamp	DeploymentId	Role	RoleInstance	CounterName	Count	Total	Minimum	Maximum
0634968969...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET Applications(__Total__)\Errors Total/Sec	2	0	0	0
0634968969...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET Applications(__Total__)\Requests/Sec	2	0	0	0
0634968969...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET\Requests Queued	2	0	0	0
0634968969...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET\Requests Rejected	2	0	0	0
0634968969...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\Memory\Available MBytes	2	2298	1147	1151
0634968969...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\Web Service(_Total)\Bytes Total/Sec	2	0	0	0
0634968969...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\Web Service(_Total)\ISAPI Extension Requests/sec	2	0	0	0
0634968972...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET Applications(__Total__)\Errors Total/Sec	2	0	0	0
0634968972...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET Applications(__Total__)\Requests/Sec	2	0	0	0
0634968972...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET\Requests Queued	2	0	0	0
0634968972...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\ASP.NET\Requests Rejected	2	0	0	0
0634968972...	ca445e...	2/19/2013...	ca445e6f3569...	csharp	csharp_IN_0	\Memory\Available MBytes	2	2457	1228	1229

FIGURE 7-10

The specific data in table 1HR is configured to store the following metrics on an hourly basis:

- ASP.NET Applications(_Total_)\Errors Total/Sec
- ASP.NET Applications(_Total_)\Requests/Sec
- ASP.NET\Requests Queued
- ASP.NET\Requests Rejected
- Memory\Available Mbytes

You use the data in the tables to create graphs and reports useful for deeper analyses into the performance and stability of the Cloud Service. For example, you can import the data into Excel and create graphs and summaries.

Configure a Cloud Service from Visual Studio

Many of the configurations you make while logged into the online Windows Azure Management Console can also be performed from Visual Studio. For example, right-click the Web Role as shown in Figure 7-11, then click Properties.

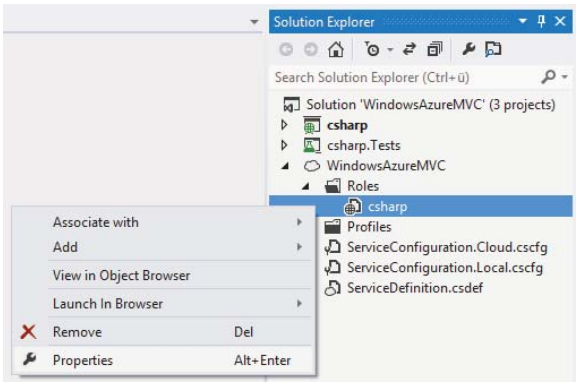


FIGURE 7-11

www.it-ebooks.info

By clicking the Properties menu item, a tab opens containing the following options also illustrated in Figure 7-12.

- **Configuration:** Supports scaling, VM Size, enabling and disabling HTTP endpoints, and diagnostics.
- **Settings:** An interface for adding settings or modifying existing settings like:
  - Diagnostics.ConnectionString
  - RemoteAccess.Enabled
  - RemoteAccess.AccountUsername
  - RemoteAccess.AccountEncryptedPassword
  - RemoteAccess.AccountExpiration
  - RemoteForwarder.Enabled
  - Caching.NamedCaches
  - Caching.DiagnosticLevel
  - Caching.CacheSizePercentage
  - Caching.ConfigStoreConnectionString

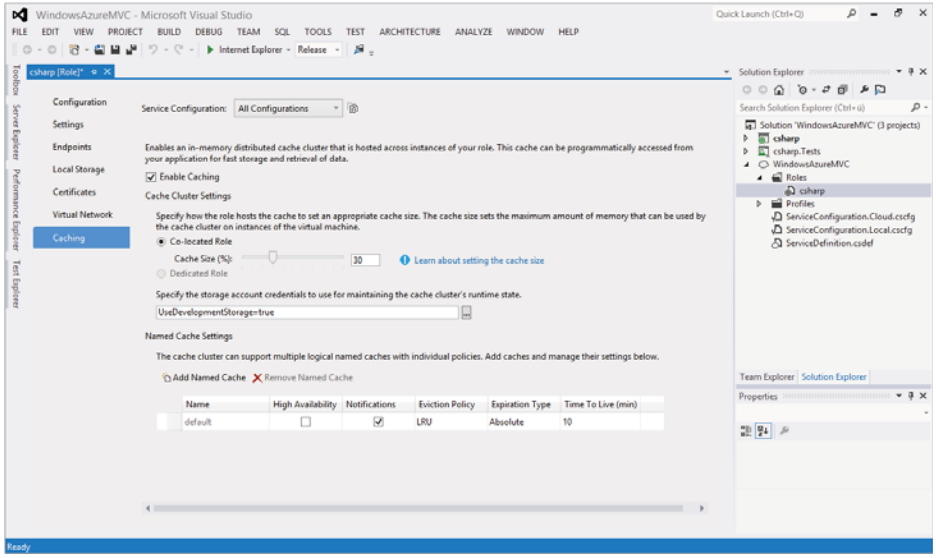


FIGURE 7-12

www.it-ebooks.info

- **Endpoints:** For adding and updating endpoints for performance monitoring.
- **Local Storage:** For adding a local storage to be used by the diagnostics capabilities.
- **Certificates.** For adding or updating certificates.
- **Virtual Networks.** Useful for management and administration of Virtual Networks hosted on the Windows Azure platform.
- **Caching.** For enabling, disabling and configuring caching.

Changes made to the configuration in Visual Studio are applied to the environment when you publish changes or deploy the Web Role to the Windows Azure Platform. Changes are currently not applied automatically.

Monitor a Cloud Service from within Visual Studio

Viewing diagnostic data in near real-time is possible from Visual Studio. As mentioned previously and illustrated in Figure 7-9, it is possible to connect to the Web Roles' storage to view and extract diagnostic data. You can find another useful Visual Studio feature on the Diagnostics summary tab. You access this tab by right-clicking the Web Role under the Windows Azure Compute feature in the Server Manager Explorer. This is illustrated with Figure 7-13.

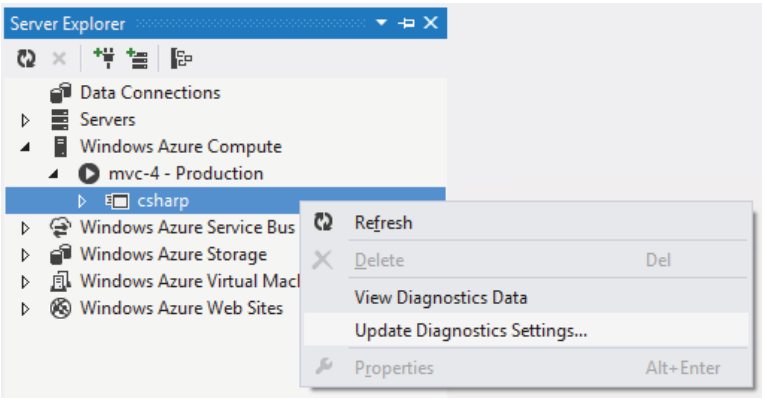


FIGURE 7-13

The Diagnostics Summary tab is rendered for the selected Web Role. The contents on the tab refresh every 15 minutes or you can refresh it manually by clicking the Refresh button. The Diagnostics summary tab is illustrated with Figure 7-14.

www.it-ebooks.info

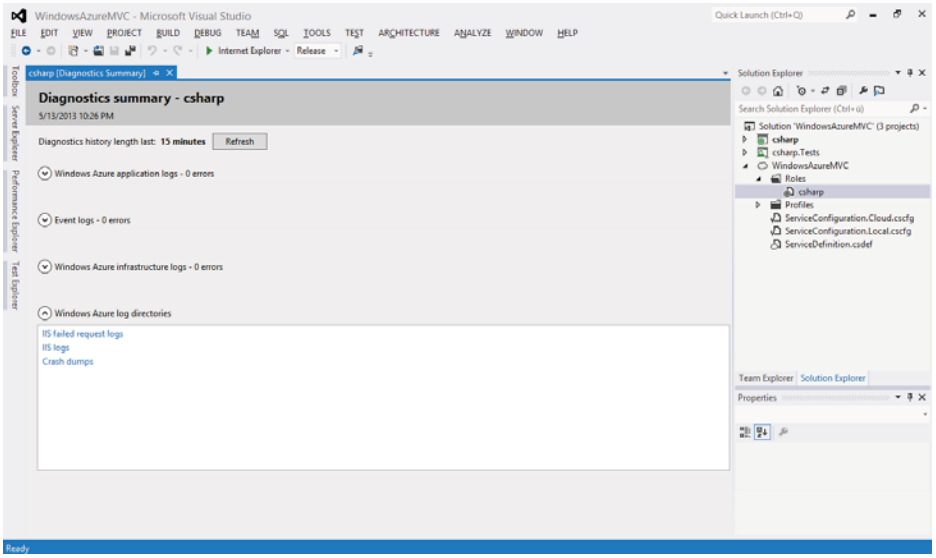


FIGURE 7-14

MANAGEMENT AND MONITORING TOOLS FOR WINDOWS AZURE

The standard tools and features available in the Windows Azure management console give administrators a valuable and clear overview of the performance, availability, and stability of their website or Web Role. However, numerous third-party tools are also available, and they give you additional capabilities for monitoring your solution on Windows Azure. Third-party tools for remote administration, computer usage monitoring, billing monitors, and storage management tools are available.

NOTE *The cloud space is changing at a rapid pace. Therefore, the links, tools, and features provided in this section may be out of date rather quickly. You can learn more about the most current tools by performing a quick Bing or Google search for* ***Windows Azure Monitoring Tools.***

Open Source Tools

The open source tools in the following list are projects that are created, driven, and supported by the community using them. Open source tools, applications, and libraries generally have no formal support. Instead, administrators get their support from online forums. In many cases, a project team manages bug changes and new feature requests so that the code-base remains stable. Because these are open source tools, companies or individuals can branch the source and implement new features or fix bugs themselves.

www.it-ebooks.info

- **Windows Azure Platform Management Tool (MMC):** Helps you easily manage your Windows Azure-hosted services and storage accounts. This tool comes as a sample with complete source code, so you can see how to perform various management and configuration tasks using the Windows Azure Management and Diagnostics APIs. For more information, see `http://wapmmc.codeplex.com/`.
- **Windows Azure Diagnostics Monitor:** A sample application that enables you to view and analyze basic Windows Azure Diagnostics information using a simple web interface. Visit: `http://archive.msdn.microsoft.com/wazdmon`.
- **Azure Application Monitor:** Helps you monitor your Azure-hosted applications in real time. It includes a library for capturing run time process information to cloud table storage, and also a desktop application for viewing the captured information in real time. For more information, see: `http://azuremonitor.codeplex.com/`.
- **Cloud Ninja:** This is a Windows Azure multitenant sample application demonstrating metering and automated scaling concepts. It also has some common multitenant features, such as automated provisioning and federated identity. For more information, see: `http://cloudninja.codeplex.com/`.
- **Cloud Samurai:** Also called Project Bowlus, this is a complete code sample demonstrating a hosted multitenant approach on Windows Azure utilizing the IIS Application Request Routing (ARR) extension. You can see this at: `http://cloudsamurai.codeplex.com/`.
- **Azure Storage Explorer:** A useful GUI tool for inspecting and altering the data in your Windows Azure Storage. Storage projects include the logs of your cloud-hosted applications. For more details, visit: `http://azurestorageexplorer.codeplex.com/`.

Windows Azure Management API

The Windows Azure Service Management API provides REST and .NET API resources that provide programmatic access to most of the functionality available from the Windows Azure management console. For example, creating a storage account, creating a cloud service, or creating and managing a virtual machine can all be performed via REST or a .NET API.

You can find more information about these capabilities at the following locations:

- **REST API Reference:** `http://msdn.microsoft.com/en-us/library/windowsazure/ee460799.aspx`
- **.NET Class Libraries:** `http://msdn.microsoft.com/en-us/library/ee393295.aspx`

The .NET Class Libraries currently have three namespaces:

- Microsoft.WindowsAzure.Diagnostics
- Microsoft.WindowsAzure.Diagnostics.Management
- Microsoft.WindowsAzure.ServerRuntime

www.it-ebooks.info

These namespaces provide a developer with a library for creating and managing diagnostics from with the application's code. For example, there is a class named `CrashDumps` that triggers the creation of a memory dump of a crashed or hung process. The memory dump can later be downloaded and analyzed to find and resolve application issues.

Windows Azure PowerShell Cmdlets

PowerShell is a powerful scripting language that Windows system administrators must know. The capabilities that PowerShell cmdlets provide continue to grow and give administrators options for managing and monitoring almost all features on the Windows platform.

NOTE *You can find information about Windows Azure PowerShell cmdlets here:* `http://msdn.microsoft.com/en-us/library/windowsazure/jj156055.aspx`.

Cmdlets fall into two broad categories:

- **For performing actions primarily in the management console.** These include those for managing your subscription, deploying and managing virtual machines, managing Virtual Networks, managing storage accounts, and deploying and managing Cloud Services.
- **For managing a SQL database hosted on the Windows Azure platform.** These provide capabilities for managing SQL server firewall rules and adding and removing a SQL server database.

Both groups support remote execution. This means you can administer your Windows Azure Web Site, Web Role, and database from a workstation. To set up PowerShell for remote execution of Windows Azure cmdlets, you need to perform the following three commands shown in Listing 7-2.

LISTING 7-2: Setting Up Remote PowerShell to Exeute Azure Cmdlets

```
PS C:\>Set-ExecutionPolicy RemoteSigned
PS C:\>Import-Module "C:\Program Files (x86)\Microsoft SDKs\
                        Windows Azure\PowerShell\
                        Azure\Azure.psd1"
PS C:\>Get-AzureSubscription
```

NOTE *If you are running on a 64-bit workstation, instead of importing the Azure.psd1 in the* `Program Files(x86)` *directory, import the one in the* `Program Files` *directory.*

---

www.it-ebooks.info

If you have multiple subscriptions, use the `Set-AzureSubscription` cmdlet so that the actions taken using PowerShell are applied to the intended subscription. There are many Windows Azure PowerShell cmdlets. To get a list of them, enter the command shown in Listing 7-3.

LISTING 7-3: Listing Windows Azure PowerShell CmdLets

```
PS C:\>get-help Azure
```

The result of the command executed in Listing 7-3 shows all existing Windows Azure PowerShell cmdlets. Any action available from the Windows Azure Management Console can be performed using a PowerShell cmdlet. These cmdlets can be used for, but not limited to:

- Managing Cloud Services
- Setting up and managing virtual machines
- Managing storage, images, and disks
- Managing certificates and SSH keys
- Creating and deleting websites

Microsoft Tools for Monitoring and Managing Windows Azure

The following list contains monitoring tools created by Microsoft. You can download and use them with that confidence that they work in tight side-by-side grouping with any existing component installed on your system.

- **System Center Monitoring Pack for Windows Azure Applications:** The Windows Azure Monitoring Pack enables you to monitor the availability and performance of applications that are running on Windows Azure.
- **The Windows Azure Management Portal:** `https://manage.windowsazure.com/`
- **Tools and Utilities Support (Windows Azure SQL Database):** A variety of tools and utilities can be used with Microsoft Windows Azure SQL Database. `http://msdn.microsoft.com/en-us/library/windowsazure/ee621784.aspx`
- **MetricsHub:** MetricsHub is a Public Cloud Monitoring service that keeps applications up and running for the lowest possible cost. `https://www.metricshub.com/`
- **Windows Azure Tools for Microsoft Visual Studio:** You can these to create, build, debug, run, and package scalable web applications and services on Windows Azure. `http://msdn.microsoft.com/en-us/library/windowsazure/ee405484.aspx`

www.it-ebooks.info

SUMMARY

In this chapter, you learned about the different monitoring options available on the Windows Azure platform. You learned some capability differences between a Windows Azure Web Site and a Web Role. For examples, although endpoints are available in both cases, you must place the website in Reserved mode for it to be configurable. Also, a Remote Desktop Connection is only possible with a Web Role. By connecting to the Web Role using Remote Desktop Connection, tools are available for troubleshooting many issues.

You also have access to numerous tools — such as PowerShell, open source, and third-party applications — for monitoring and remotely managing an application hosted on the Windows Azure platform. Chapter 8 contains exercises that illustrate many of the concepts discussed in this chapter.

www.it-ebooks.info

8 Monitoring and Supporting an ASP.NET MVC 4 Project on Windows Azure

EXERCISES AND EXAMPLES

IN THIS CHAPTER

- How to monitor and support an ASP.NET MVC 4 Web Site on Windows Azure
- How to monitor and support an ASP.NET MVC 4 Cloud Service on Windows Azure

WROX.COM CODE DOWNLOADS FOR THIS CHAPTER

You can find the wrox.com code downloads for this chapter at `www.wrox.com/go/azureaspmvcmigration` on the Download Code tab. The code for this chapter is divided into the following major examples. It is recommended that you download the code and review it so that you have a good understanding of what is about to be discussed.

Now that you're live in production with an ASP.NET MVC 4 Web Site or Cloud Service, you certainly want to keep it running and responding to the visitors to the site.

A number of problems can happen, each one requiring a different set of logs or information to resolve them. This chapter walks you through monitoring and troubleshooting, with exercises divided between the ASP.NET MVC 4 Web Site and Cloud Service.

MONITORING AND SUPPORTING AN ASP.NET MVC 4 WEB SITE ON WINDOWS AZURE

You have many monitoring capabilities available to you for both websites and Cloud Services on the Windows Azure platform. Features that are very useful for monitoring your website include:

- The performance and utilization graphs available via the Dashboard view in the Windows Azure management console
- The real-time streaming of diagnostic tracing into Visual Studio

www.it-ebooks.info

- Capabilities you can extend to include additional metrics, such as Failed Requests, Requests per Second and Memory utilization.

For the last item, you can extend capabilities from the Windows Azure management console, from Visual Studio and, for a Cloud Service, through a Remote Desktop Connection. You can then capture performance data just as you would from any web server.

These monitoring and performance capabilities are critical for understanding how a website performs from a response, availability, and stability perspective because they provide the administrator with useful data for optimizing, scaling, or repairing the system. Most inefficiencies that exist in a website would go unnoticed without monitoring performance, capturing performance data and reviewing the captured performance metrics.

Accessing the Graph and Usage Overview

One of the most useful tools for monitoring the usage, stability, and availability of your Windows Azure Web Site is the graph and usage overview features available through the Web Site Dashboard. You can access the Dashboard by clicking Web Sites ➪ mvc-4 (the website you created) ➪ Dashboard (default selection). Figure 8-1 represents how this may look.

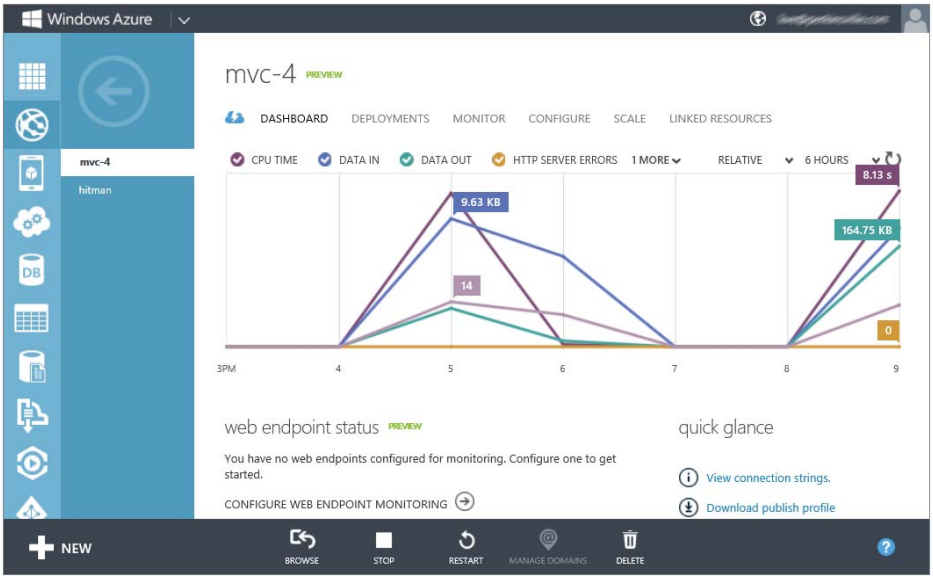


FIGURE 8-1

Pay special attention to the HTTP Server Errors attribute. If you receive any number of these errors, you may want to check the log files to identify the error and to determine from which request it originates. Information on how to download and analyze the log files are discussed in the section titled "Downloading and Analyzing Diagnostic Logs," later in this chapter.

www.it-ebooks.info

A review and logging of the other attributes, such as CPU Time, Data In, Data Out, and Requests are also useful because they identify trends related to your visitors accessing your website, and they give you a good understanding of how your system works and uses the available resources. By looking at these logs and charts over time, you gain an understanding of website's normal behavior and can quickly spot when behavior is abnormal.

By default, only a number of parameters are tracked:

- **CPU Time:** The amount of time deducted from your allocated CPU usage
- **Requests:** The number of requests sent to the website
- **Data Out:** The amount of data sent from the server to the client
- **Data In:** The amount of data sent to the server from the client
- **HTTP Server Errors:** A count of the number of HTTP server errors

Adding Metrics to the List of Monitored Attributes

Besides the default metrics listed in the last section, you can add a number of other metrics to tracking your site. To enable additional metrics, perform the following steps:

1. Select the Add Metrics link at the bottom of the Monitor page, as shown in Figure 8-2.

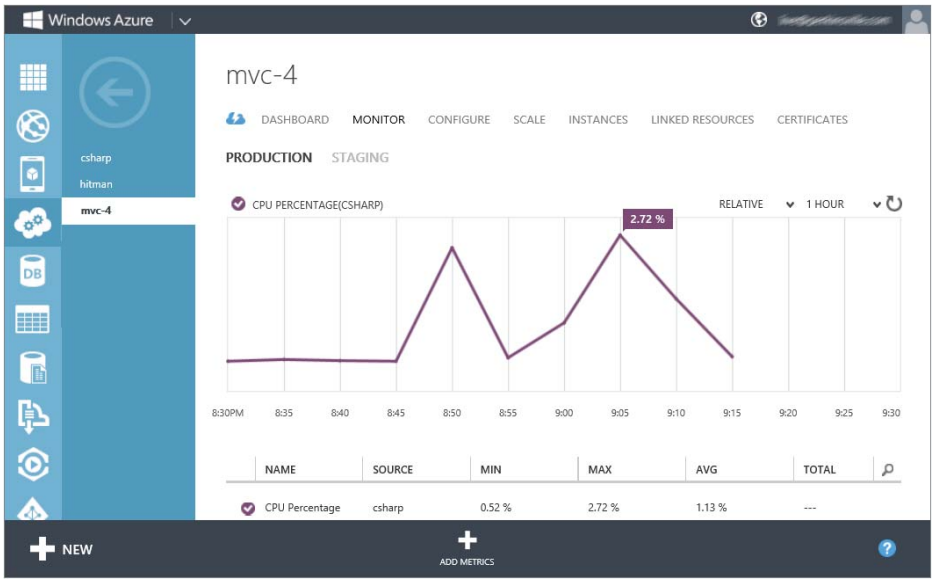


FIGURE 8-2

A pop-up appears showing additional traceable properties, which are listed in Table 8-1. Figure 8-3 illustrates the window allowing the addition or modification of metrics to track.

www.it-ebooks.info

NOTE *For a detailed discussion of the properties in Table 8-1, see Chapter 7.*

TABLE 8-1: Windows Azure Monitor Properties

NAME	UNIT	DESCRIPTION
CPU Time	Milliseconds	Amount of time the CPU is used
Data In	Bytes	Amount of data from client to server
Data Out	Bytes	Amount of data from server to client
HTTP 401 errors	Count	Authentication errors
HTTP 403 errors	Count	Forbidden errors
HTTP 404 errors	Count	Not Found errors
HTTP 406 errors	Count	Not Acceptable errors
HTTP Client errors	Count	Client resets or failures
HTTP Redirects	Count	300 HTTP status codes
HTTP Server Errors	Count	500 HTTP status codes
HTTP Successes	Count	100–200 status codes
Requests	Count	The numbers of requests sent to the server

2. Select the options you want to view on the Monitor graph, and click the check mark at the bottom of the pop-up window.
3. After focus returns to the Monitor Dashboard, scroll down to view the values of the selected metrics.

Configuring Diagnostics for the Website

Another very interesting set of website capabilities are the features located on the Configure console. These features include the items — such as application diagnostics, site diagnostics, and deployment settings — that all provide useful information for troubleshooting and monitoring your website. To configure diagnostics for your website, perform the following:

1. Click the Configure link at the top of the Windows Azure management console for the given website.
2. Scroll down the page until you come to the site diagnostics section, as shown in Figure 8-4.

www.it-ebooks.info

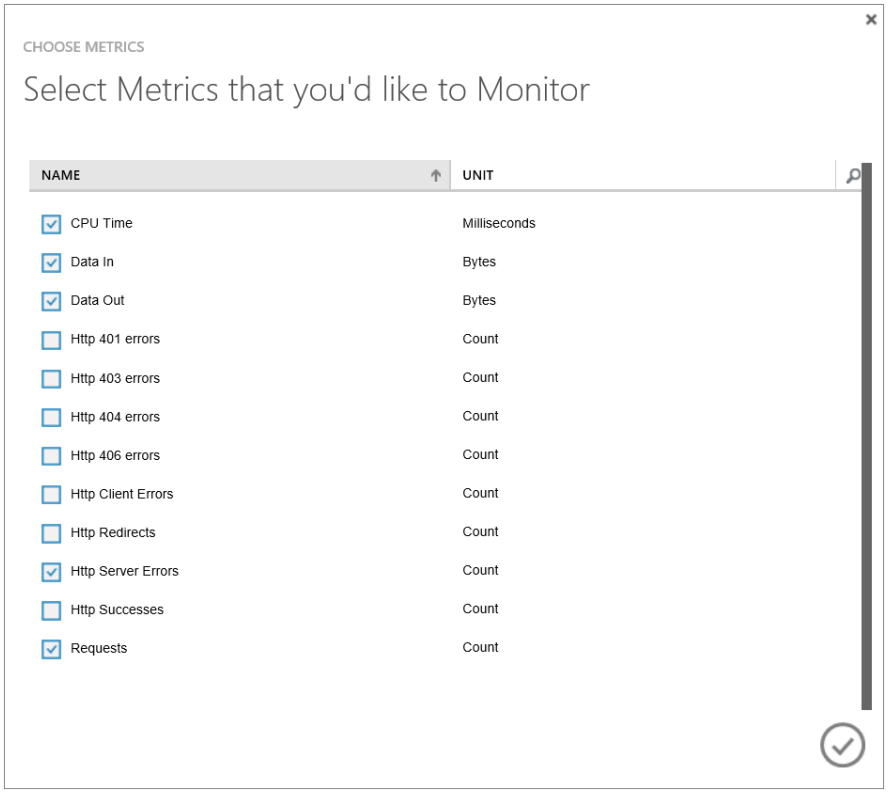


FIGURE 8-3

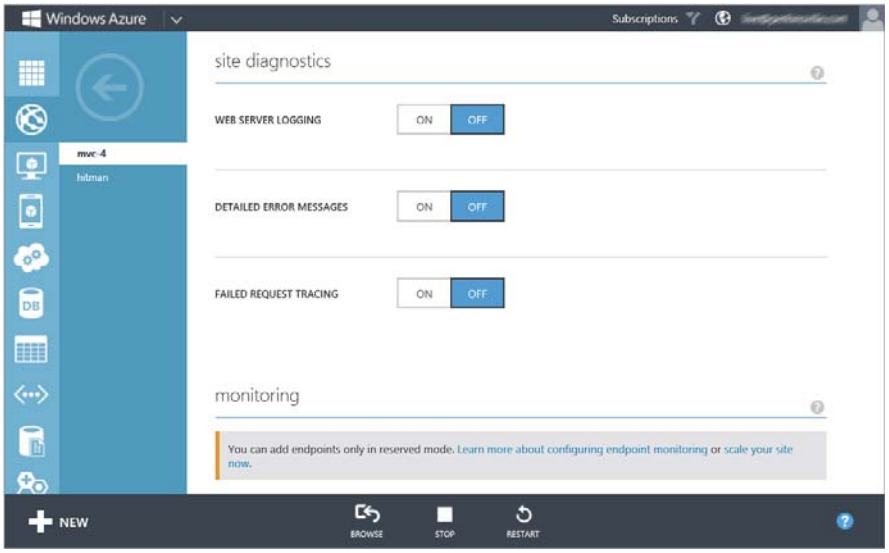


FIGURE 8-4

www.it-ebooks.info

The following diagnostic logs, provided in Table 8-2, are available for logging and are off by default.

TABLE 8-2: Site Diagnostic Features

NAME	DESCRIPTION
Web Server Logging	IIS logs containing requests and HTTP status codes
Detailed Error Messages	Requests resulting in an error saved as an HTM file
Failed Request Tracing	Detailed report of the HTTP requests flowing through the request pipeline

3. For this example, turn all the logging on, as shown in Figure 8-5, then select the Save button at the bottom of the page.

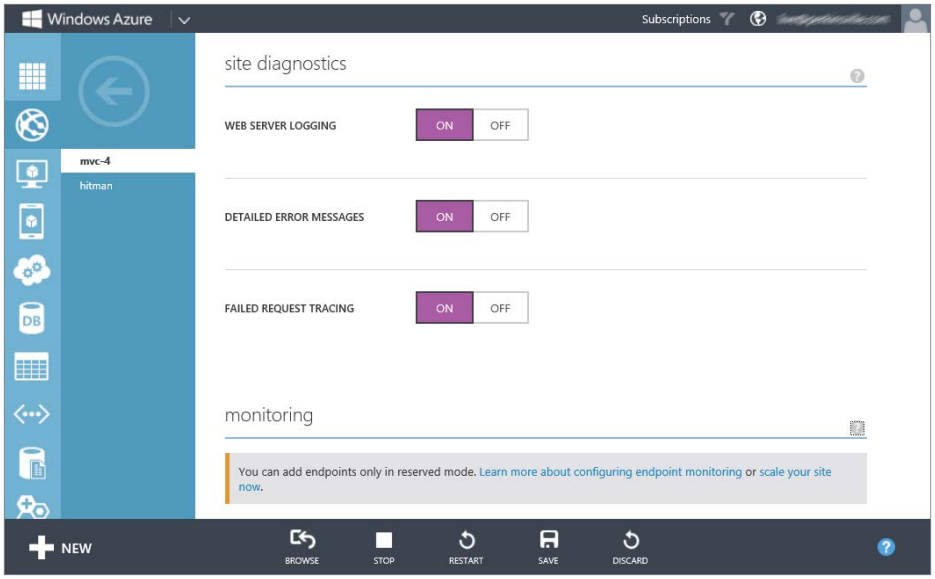


FIGURE 8-5

4. When successfully configured and saved, access the website so that some traffic is logged and let it run for some minutes.
5. Proceed to the next exercise, which walks you through downloading and reviewing the captured logs.

www.it-ebooks.info

Downloading and Analyzing Diagnostic Logs

Now that the website is configured to begin logging web server, detailed error, and failed request logs, you need to download these logs so you can analyze them and take actions to prevent errors or to improve website performance, if required. To download and analyze the just-configured diagnostic logs, perform the following steps:

1. Scroll down on the Dashboard console. On the right side of the page in the Quick Glance section, locate the Diagnostics Log heading. Under that heading is an FTP address that you can use to download the log files.
2. Obtain a download tool, such as FileZilla, which you can download here: `http://filezilla-project.org/download.php`.
3. Once you download and install an FTP client, use the FTP address and the UID/Password you created previously in Chapter 6 to access the FTP site.

NOTE *An example of how the FileZilla client may look when connected to the LogFiles directory of the Windows Azure Web Site is shown in Figure 8-6.*

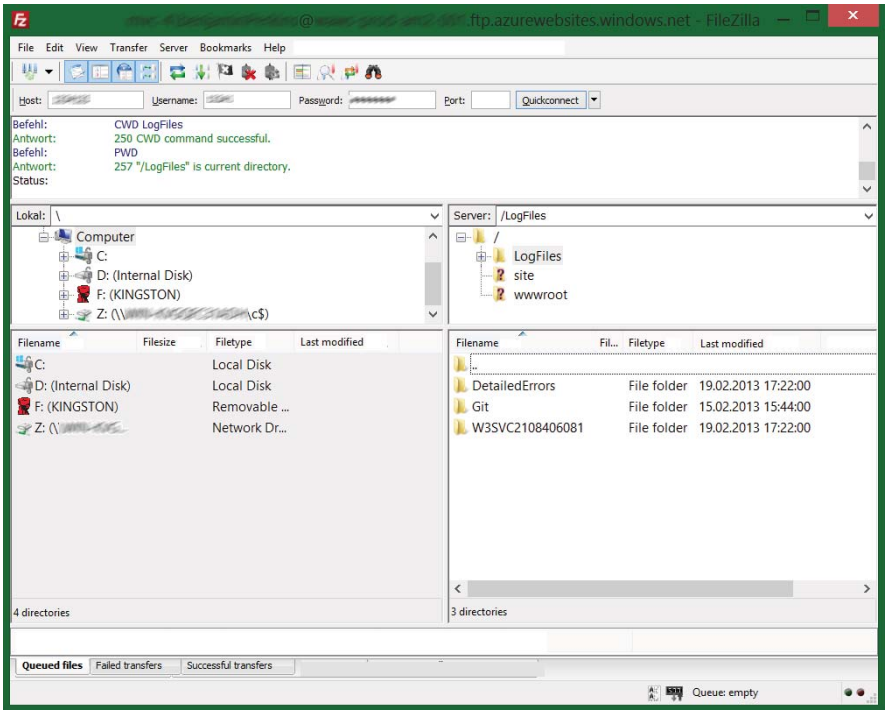


FIGURE 8-6

www.it-ebooks.info

4. Locate two directories in Figure 8-6: DetailedErrors and W3SVC*, where * is the website ID.
5. Click the DetailedErrors directory. This will open a file called ErrorPage****.htm. Download one or all of the Error Pages and view them. For example, at first glance, Figure 8-7 shows that a visitor attempted to access a file that was not found, that is, a 404.0 Not Found error.
6. Look at the Requested URL value in the Detailed Error Information section. The requested file is robot.txt. There is a lot of valuable information on the page to read.

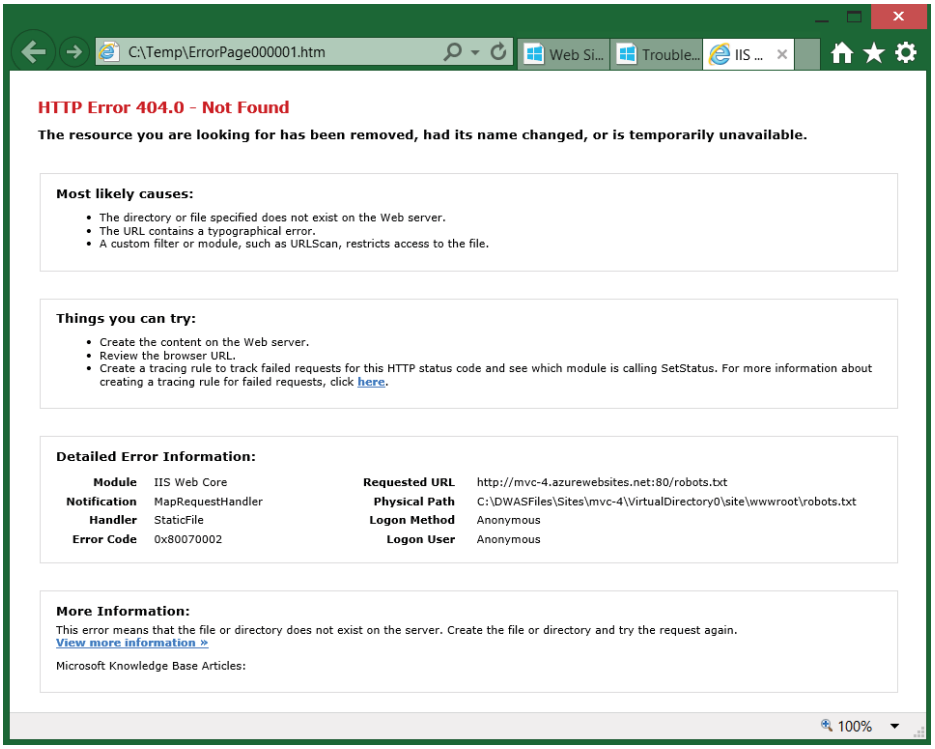

C:\Temp\ErrorPage000001.htm
Web Si...
Trouble...
IIS ...
HTTP Error 404.0 - Not Found
The resource you are looking for has been removed, had its name changed, or is temporarily unavailable.
Most likely causes:
The directory or file specified does not exist on the Web server.
The URL contains a typographical error.
A custom filter or module, such as URLScan, restricts access to the file.
Things you can try:
Create the content on the Web server.
Review the browser URL.
Create a tracing rule to track failed requests for this HTTP status code and see which module is calling SetStatus. For more information about creating a tracing rule for failed requests, click here.
Detailed Error Information:
Module IIS Web Core
Notification MapRequestHandler
Handler StaticFile
Error Code 0x80070002
Requested URL http://mvc-4.azurewebsites.net:80/robots.txt
Physical Path C:\DWASFiles\Sites\mvc-4\VirtualDirectory0\site\wwwroot\robots.txt
Logon Method Anonymous
Logon User Anonymous
More Information:
This error means that the file or directory does not exist on the server. Create the file or directory and try the request again.
View more information »
Microsoft Knowledge Base Articles:
100%

FIGURE 8-7

7. If the error is preventable based on some action, execute that action.

NOTE *The error page is the same that you receive when an error occurs on a normal instance of IIS 7+.*

www.it-ebooks.info

In this example, there was a request to a robots.txt file that does not exist on the website. The robot.txt file tells compliant web robots which files or directories it should exclude from being categorized or archived into a search engine. If there is no robots.txt file, then all contents of the website is crawled and cataloged. If there is a section of your public website that shouldn't render in any search engine or be archived, you must place a text file in the root directory of the website.

- If you do not want anything in the Images directory cataloged or archived, the robot.txt file would resemble Listing 8-1.

LISTING 8-1: Restricting Access to All Robots and Crawlers to the Images Directory

```
User-agent: *
Disallow: /Images/
```

- Listing 8-2 is an example of a robot.txt file that restricts access to a specific file if you prefer to restrict access on a file basis.

LISTING 8-2: Restricting Access to All Robots and Crawlers to a Specific File

```
User-agent: *
Disallow: /Images/private.png
```

Even though a robot.txt exists that disallows access to some files and directories, it is up to the crawler to honor the instructions.

8. Click the W3SVC* directory (where * is the website ID IIS created when the site was configured). This will show the Failed Request Tracing logs.
9. Download all or a sample of the XML files and the freb.xsl file. The XSL file is required for formatting the output of the Failed Request Tracing XML file.
10. Open the XML file in a browser such as Internet Explorer, and you should see something similar to Figure 8-8.
11. You can find an abundant amount of information in this log. For example, click the Request Details tab ➪ Performance View. As shown in Figure 8-9, you see the ASP.NET MVC 4 request pipeline events. On the right side are timings. If you were experiencing a performance issue, this would be a good place to determine what the problem is.
12. Click the Compact View tab, as shown in Figure 8-10. This tab shows a line-by-line breakdown of what happened from the beginning of your request to the final response. This is not only good for finding issues, but is also a great source for learning how an ASP.NET MVC 4 request flows through the IIS pipeline.

www.it-ebooks.info

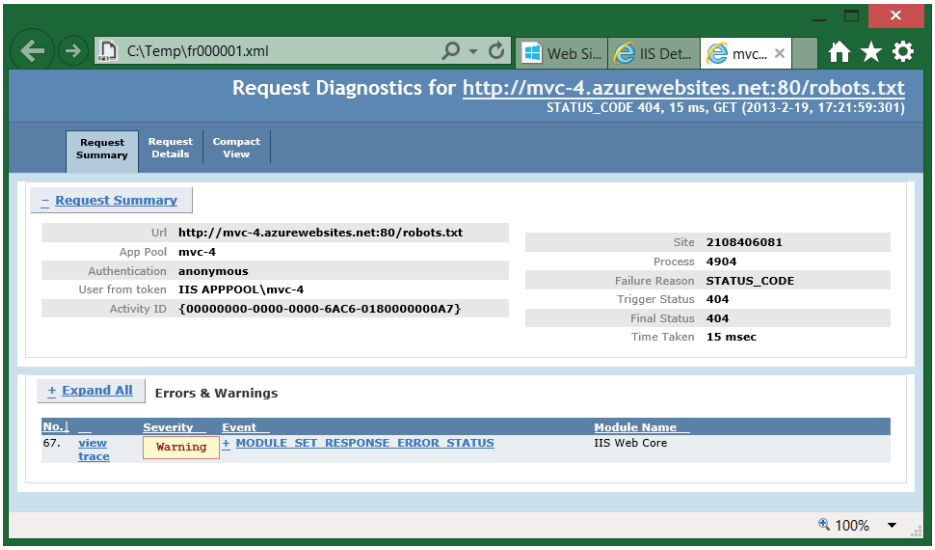


FIGURE 8-8

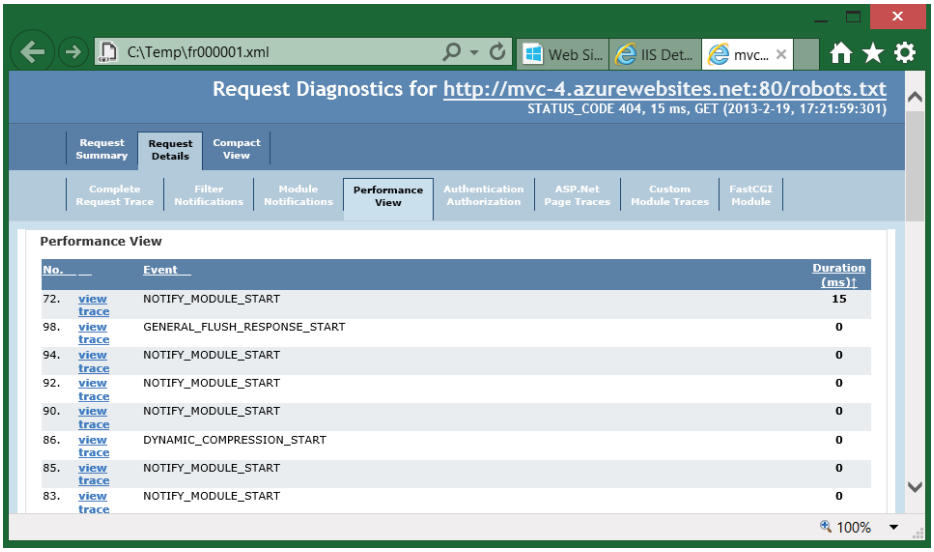


FIGURE 8-9

Streaming Diagnostic Logs

One of the newer capabilities supports streaming Windows Azure Web Site logs directly into Visual Studio. This information is useful for debugging the application in real-time while users

www.it-ebooks.info

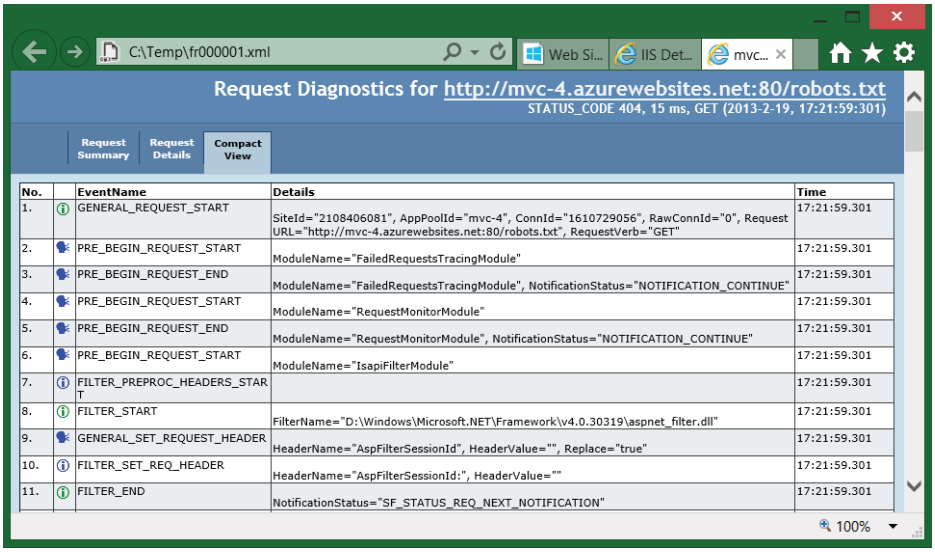


FIGURE 8-10

are accessing and using the Site. To implement diagnostic log streaming, perform the following steps:

1. **Add a Trace statement to the ASP.NET MVC 4 Web Site project.** Open the sample MVC project in Visual Studio and open the Controllers\HomeController.cs file.

2 **Add code.** Add the code shown in Listing 8-3 to the existing `Index()` method in the HomeController.cs file.

LISTING 8-3: Adding a Trace statement to the Index() method

```
using System.Diagnostics;

public ActionResult Index()
{
  Trace.TraceError("An error happened here is the message: Just testing! ");
}
```

3. **Publish the changes.** You have numerous ways to publish the code change to the Windows Azure Web Site. See Chapter 5 or 6 for information about these different deployment methods. Once the modified Controller\HomeController.cs file is deployed, continue to step 4.

4. **Enable the application logging.** To enable the logging, open the Server Explorer in Visual Studio as shown in Figure 8-11 and expand the Windows Azure Web Sites feature.

www.it-ebooks.info

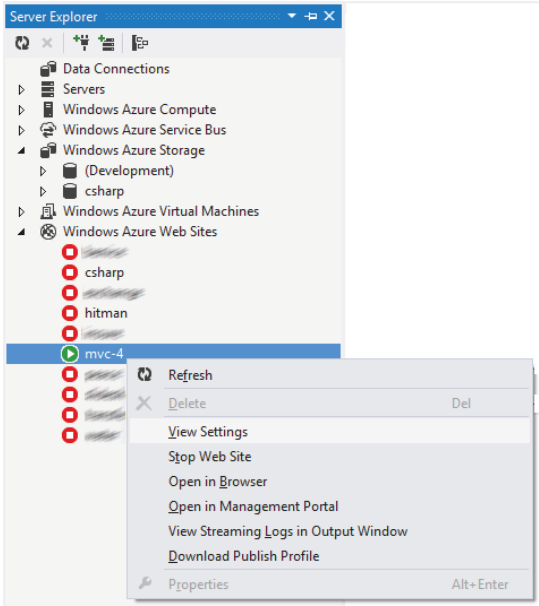


FIGURE 8-11

5. **Open the website settings.** Right-click the website to which the Trace statement was added in step 2. In this example, right-click the mvc-4 website and click View Settings.

6. **View the settings.** Clicking the View Settings menu item opens a properties page similar to that shown in Figure 8-12.

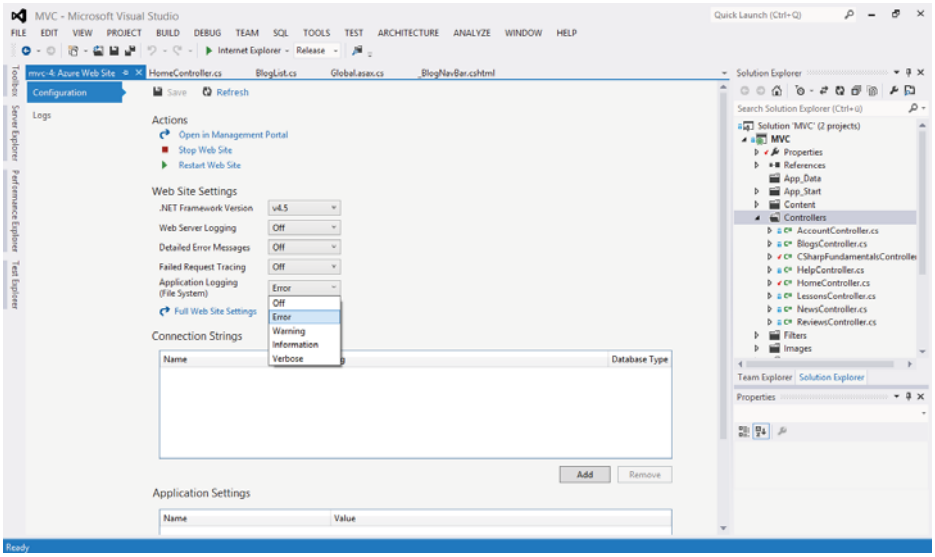


FIGURE 8-12

www.it-ebooks.info

7. **Change the settings.** Set the Application Logging (File System) value to Error and click the Save link on the top left side of the properties page.
8. **View the streaming logs in the output window.** Once the setting has been saved, return to the Server Explorer and right-click the website similar to what was done in step 5.
9. **View the streaming diagnostics.** Right-clicking the website brings up a pop-up menu, where you must select the View Streaming Logs in Output Window. This opens an Output window where you can view the streaming diagnostics in almost real-time as illustrated in Figure 8-13. Note that there is about a two second delay.

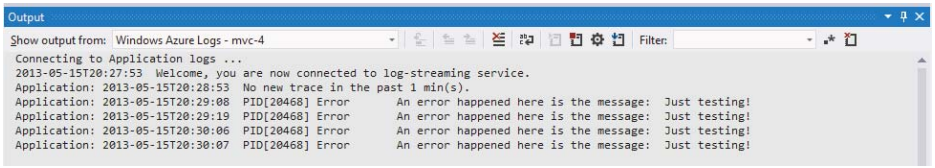

Output
Show output from: Windows Azure Logs - mvc-4
Filter:
Connecting to Application logs ...
2013-05-15T20:27:53 Welcome, you are now connected to log-streaming service.
Application: 2013-05-15T20:28:53 No new trace in the past 1 min(s).
Application: 2013-05-15T20:29:08 PID[20468] Error An error happened here is the message: Just testing!
Application: 2013-05-15T20:29:19 PID[20468] Error An error happened here is the message: Just testing!
Application: 2013-05-15T20:30:06 PID[20468] Error An error happened here is the message: Just testing!
Application: 2013-05-15T20:30:07 PID[20468] Error An error happened here is the message: Just testing!

FIGURE 8-13

10. **Stop streaming the diagnostic logs.** Once you have viewed enough of the streaming diagnostic data, turn it off by right-clicking the website and selecting the Stop Viewing Logs menu item.

Now that you know some details about configuring the monitoring for a website, you can move to the next section to check out what is available with a Cloud Service.

MONITORING AND SUPPORTING AN ASP.NET MVC 4 CLOUD SERVICE ON WINDOWS AZURE

As previously discussed, there are a number of differences between a website and a Cloud Server and Web Role. One is the ability to update diagnostics settings on a live service, and the other is to create a Remote Desktop Connection. The steps described in the following sections show you how to set up both these features.

Updating Diagnostic Settings on a Live Service

You can configure the metrics that track the performance and availability of a Web Role from the Windows Azure management console. In addition, the recent release of the Windows Azure SDK now allows you to perform that same configuration from Visual Studio. To update the diagnostic settings for a Cloud Service hosted on the Windows Azure platform using Visual Studio, perform the following steps:

www.it-ebooks.info

1. **Update the Diagnostic settings.** Open the Server Explorer menu, expand the Windows Azure Compute feature, and right-click the Update Diagnostics Settings menu item as shown in Figure 8-14. Then click the Update Diagnostics Settings menu item to open the window shown in Figure 8-15.

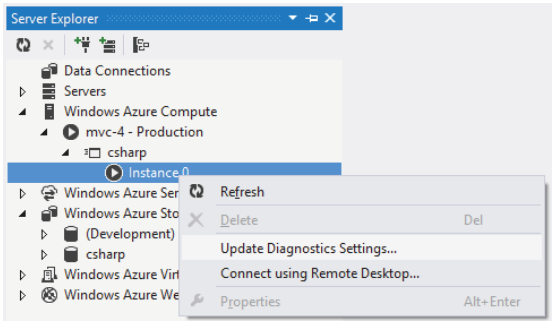


FIGURE 8-14

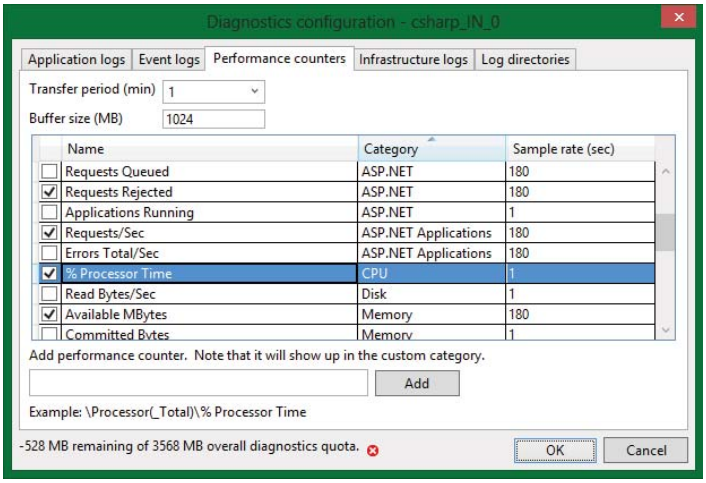


FIGURE 8-15

2. **Configure and Enable Performance Counters.** Select the desired metrics from the numerous performance counters and click OK to immediately begin logging those metrics.

3. **View the data stored in the Windows Azure Storage table.** The performance data is stored in the linked Windows Azure Storage account. To access the table, open Server Explore in Visual Studio as shown in Figure 8-16.

www.it-ebooks.info

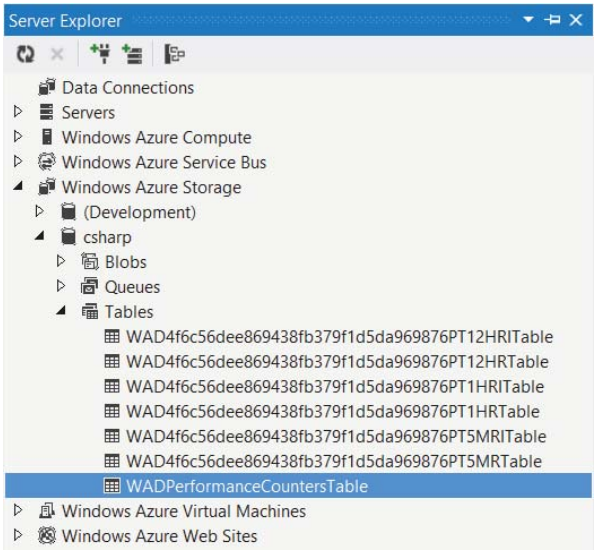


FIGURE 8-16

4. **Extract data.** Double-click the highlighted table called WADPerformanceCountersTable. The table contents appears in the main window of Visual Studio. You can now make a data connection to the table for extracting and analysis offline.

The information logged in the WADPerformanceCountersTable is useful for forecasting and measuring resource usage. You can create action plans to scale and make code enhancements by analyzing this data.

NOTE *Recall from Chapter 6 that the ASP.NET MVC 4 project called mvc-4 was converted to an ASP.NET MVC 4 Web Role named csharp. You need to know this when you perform the steps in the next section "Configuring a Remote Desktop Connection."*

Configuring a Remote Desktop Connection

You may find it useful to create a Remote Desktop Connection when you want to troubleshoot or monitor a Web Role hosted on the Windows Azure platform.

If you have not already published your ASP.NET MVC 4 project, you can configure the Remote Desktop configuration during the initial publishing of the Web Role. If you have already published your ASP.NET MVC 4 project and want to set up only the Remote Desktop Connection without republishing, go to the next section titled "Setting Up a Remote Desktop Connection for an Existing Cloud Service."

www.it-ebooks.info

To begin the configuration of a Remote Desktop Connection to a Web Role, perform the following:

1. **Publish the Web Role with the Enable the Remote Desktop for All Roles option enabled.** Right-click the project name, in this case csharp, and click Publish, as shown in Figure 8-17.

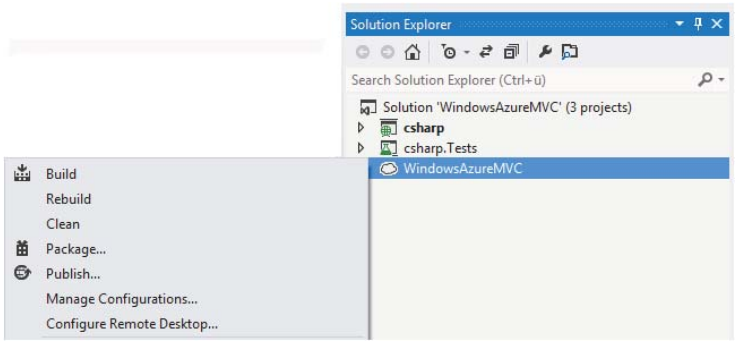


FIGURE 8-17

2. **Click Publish to start the Navigation Wizard.** This is shown in Figure 8-18. If you do not already have a credential file, click the Sign in to Download Credentials link to retrieve it.

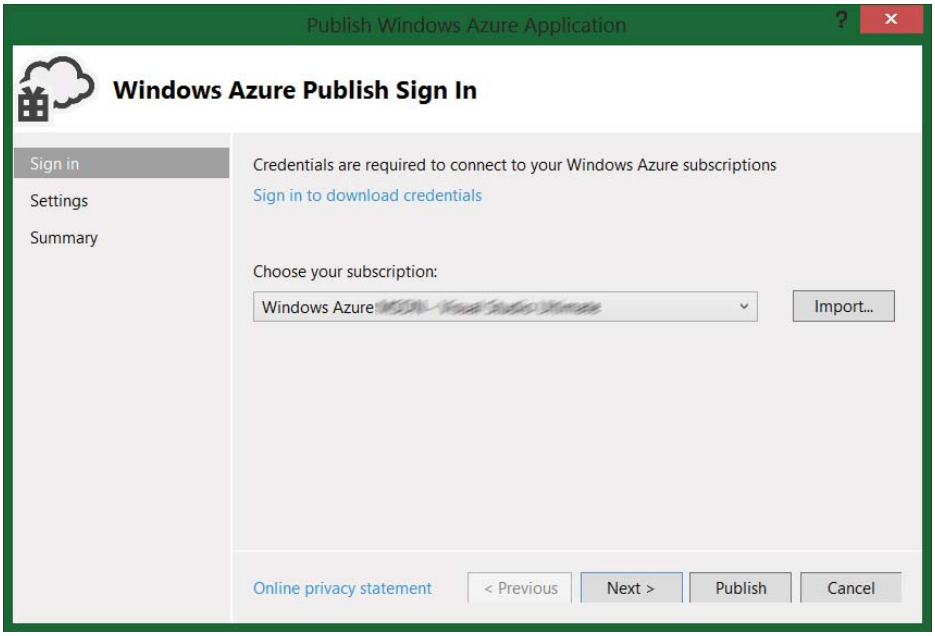


FIGURE 8-18

www.it-ebooks.info

3. **Download the publish file.** When you log into your Windows Azure account, a file with the extension .publishingsettings is downloaded. Save the file to a location on your computer; then click the Import button and navigate to and select the file just downloaded.

4. **Open the Windows Azure Publish Settings Window.** When the publishing settings file is successfully imported, click the Next button. The next window in the wizard displays, as shown in Figure 8-19.

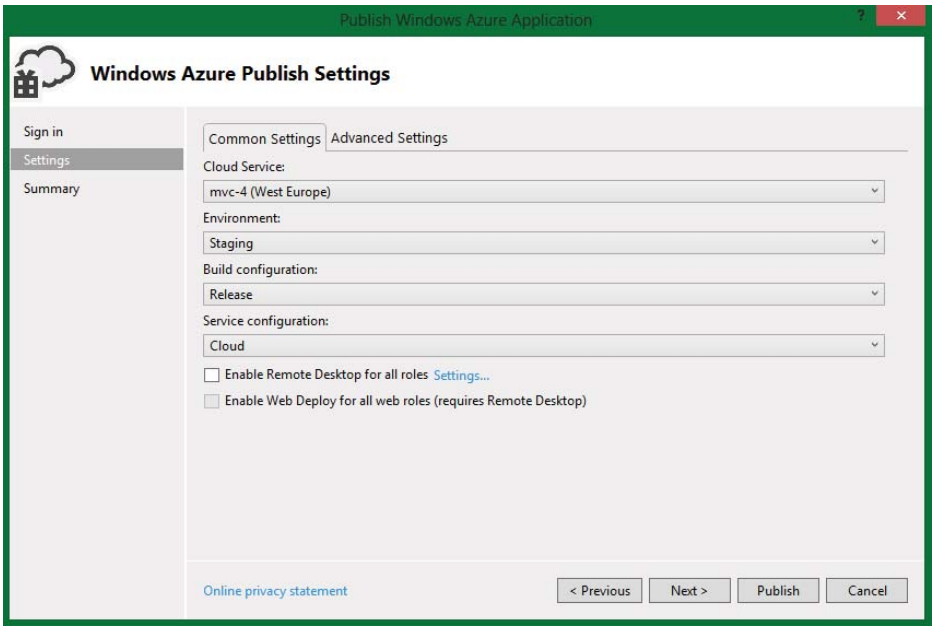


FIGURE 8-19

5. **Enable the Remote Desktop.** By default, the Enable Remote Desktop for All Roles Settings is not selected. Select the check box to enable the feature.

6. **Provide the User ID, Password, and certificate for making the connection.** When you click the Enable Remote Desktop for All Roles check box, a pop-up window prompts you for a username, password, and an expiration date.

7. **Choose a strong password that is not the same as the username, and set a reasonable account expiration date.** Figure 8-20 illustrates this. The strength of the certificate you select for password encryption in the coming steps, as well as the complexity of your password, should dictate the duration of this accounts validity — that is, the expiration date. Like all account credentials, you should change this credential regularly to avoid a breach in security.

www.it-ebooks.info

8. **Enter in the other requested information in Figure 8-20.** If you want to select or create a certificate to encrypt the user credentials, select the More Options button. Without selecting an encryption certificate, a certificate is automatically created and used. When complete, click OK.

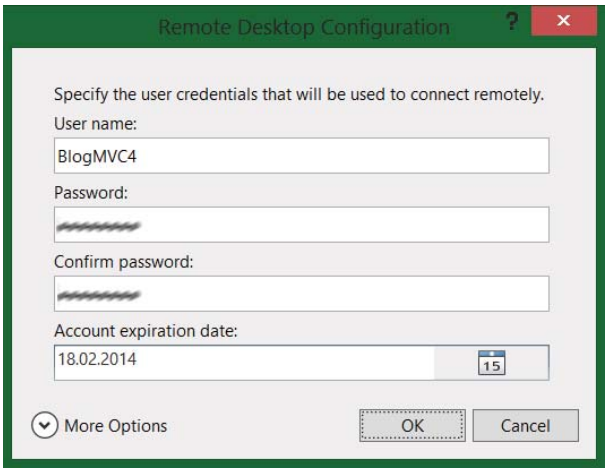


FIGURE 8-20

9. **View the Publish Summary.** Select the Next button to view the Summary. Confirm that the Remote Desktop attribute has a value of Enabled, as shown in Figure 8-21.

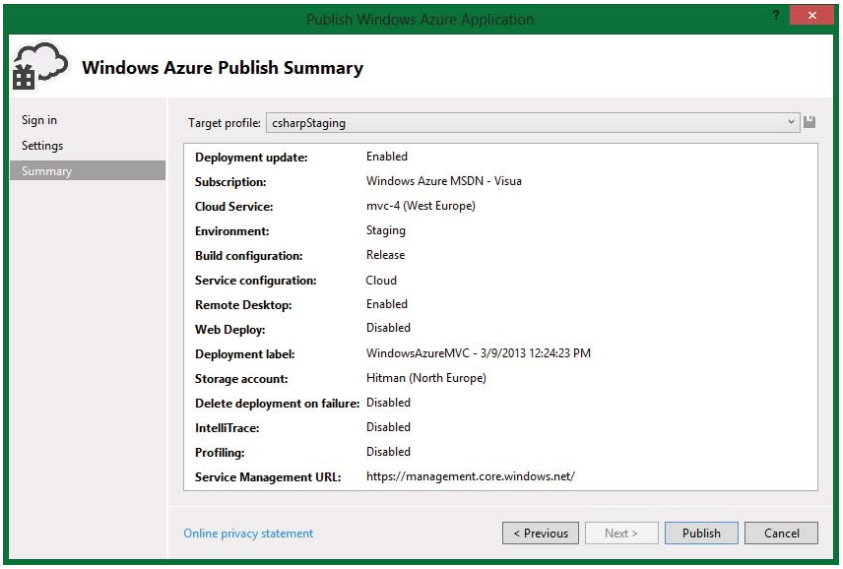


FIGURE 8-21

www.it-ebooks.info

11. **Publish the Web Role.** When confirmed, select the Publish button to publish the Web Role and to enable the Remote Desktop. A status window displays when the Role publishes. Figure 8-22 illustrates how the status appears during the publishing, and when it completes.

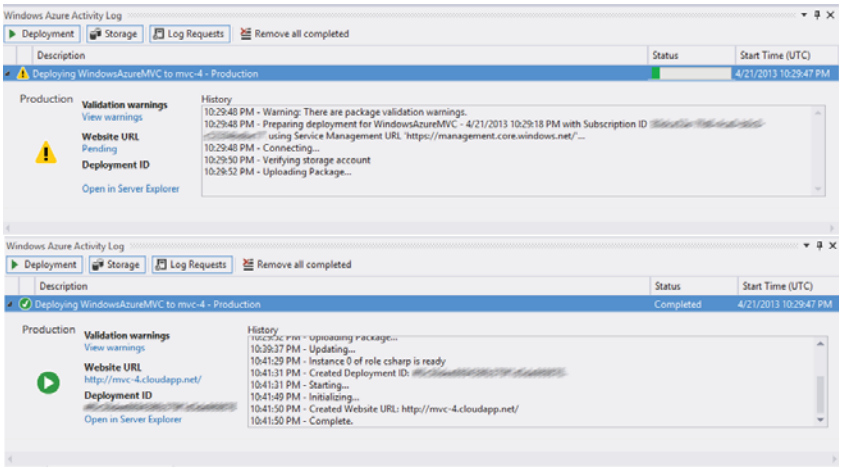


FIGURE 8-22

12. **Connect the Web Role.** When the publishing is complete, log in to the Windows Azure management console, and navigate to the Web Role that was just published. Click the Instances option as shown in Figure 8-23 and select the Connection link.

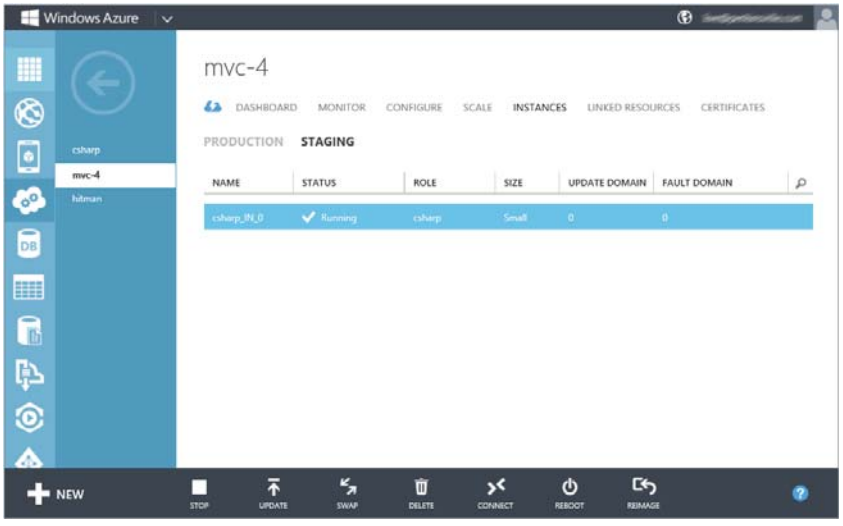


FIGURE 8-23

www.it-ebooks.info

13. **Download, save and open the RDP file for making the Remote Desktop Connection.** The RDP file is a configuration file used with the Remote Desktop Connection application, which contains the information required to make the connection to the remote server. When you click the RDP file, a pop-up displays, as shown in Figure 8-24. Click the Open button to open the Remote Desktop Connection and connect immediately.

NOTE *Alternatively, you can select the Save button and save the link to your desktop or any location on your computer. By saving the connection you have the option to make a connection again in the future without having to access the Windows Azure management console.*

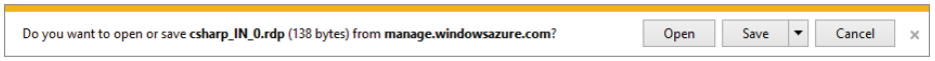


FIGURE 8-24

14. **Enter your credientials.** Whether you open or download and save the file, double-click the icon, and enter the credentials you provided in Figure 8-20. Then select OK. Figure 8-25 provides an example of this window.

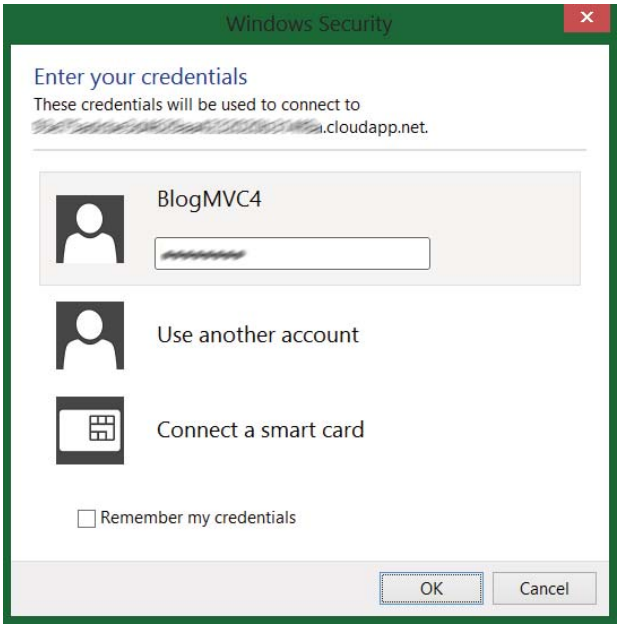


FIGURE 8-25

www.it-ebooks.info

15. **Connect to the Web Role, make some changes, and disconnect from the Web Role.** After logging in. you are given control of the server; for example, if the server is Windows Server 2012, you see a desktop similar to that shown in Figure 8-26.

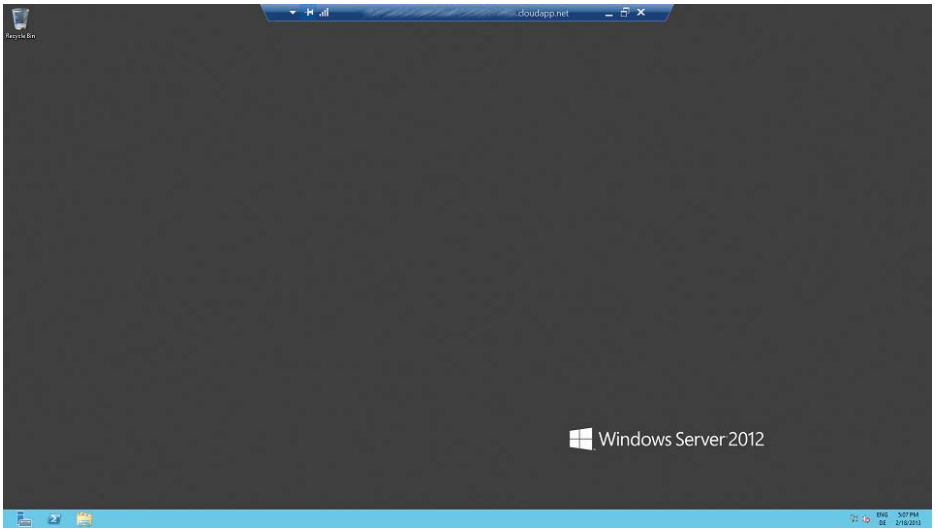


FIGURE 8-26

On the desktop, you can view files and modify the server or IIS configuration as you would with any server. To end the session, simply log out like you would on any server and you are disconnected.

Setting Up a Remote Desktop Connection for an Existing Cloud Service

If you have already published your Web Role and do not want or need to do a republish to set up a Remote Desktop Connection, you can achieve the same connection by performing the steps in this section. The steps to configure a Remote Desktop Connection previously required that you publish the configuration file to the Windows Azure platform. Using the following approach, no publishing is required and you can perform the configuration either directly in Visual Studio or the Windows Azure management console.

NOTE *You must be connected to the Internet for the exercises in this chapter to work as expected.*

---

www.it-ebooks.info

1. Right-click the project (not the Web Role) and then click Configure Remote Desktop, as shown in Figure 8-27. The Remote Desktop Configuration window appears, as shown in Figure 8-28.

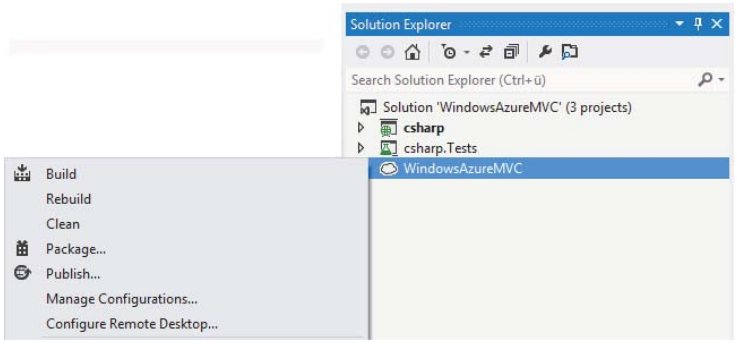


FIGURE 8-27

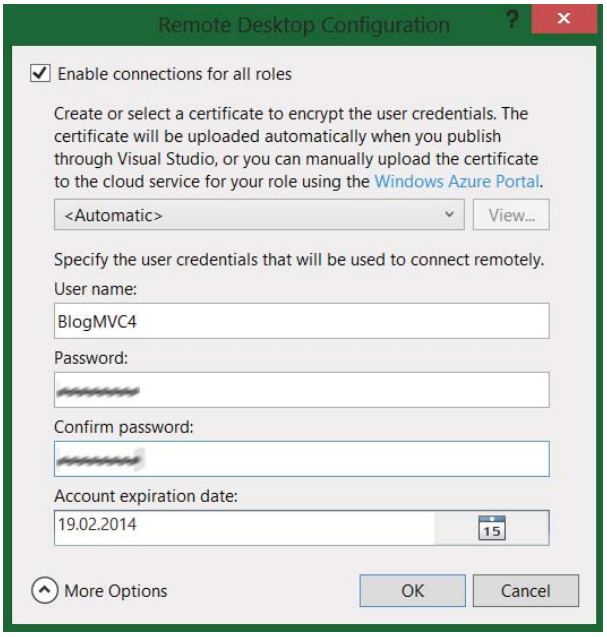

Remote Desktop Configuration
Enable connections for all roles
Create or select a certificate to encrypt the user credentials. The certificate will be uploaded automatically when you publish through Visual Studio, or you can manually upload the certificate to the cloud service for your role using the Windows Azure Portal.
<Automatic>
View...
Specify the user credentials that will be used to connect remotely.
User name:
BlogMVC4
Password:
Confirm password:
Account expiration date:
19.02.2014
More Options
OK
Cancel

FIGURE 8-28

2. Select the Enable Connections for All Roles check box. Then select a certificate to use for the encryption of the user credentials (or leave at <Automatic>).
3. Enter the username, password, and expiration date.
4. Click OK after entering the required information.

www.it-ebooks.info

5. Log in to the Windows Azure management console. Then select Web Sites ➪ mvc-4 (your Web Role) ➪ Instances. Notice that the Connect icon is enabled, as shown in Figure 8-29.

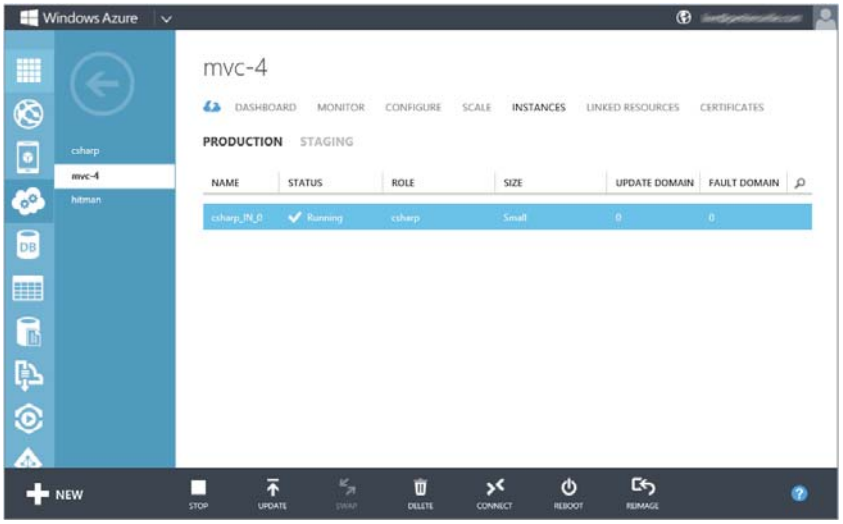


FIGURE 8-29

6. Click the Connect link, and you are prompted to open or save the RDP file, as shown in Figure 8-30.

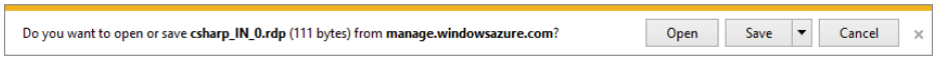


FIGURE 8-30

7. Click the Open button, and the pop-up window shown in Figure 8-31 displays.
8. Click the Connect button to continue.

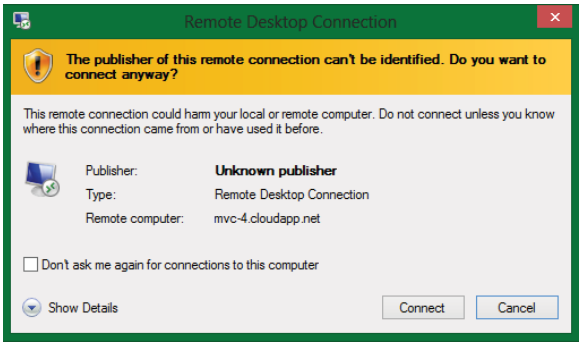

Remote Desktop Connection
The publisher of this remote connection can't be identified. Do you want to connect anyway?
This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.
Publisher:
Unknown publisher
Type:
Remote Desktop Connection
Remote computer:
mvc-4.cloudapp.net
Don't ask me again for connections to this computer
Show Details
Connect
Cancel

FIGURE 8-31

www.it-ebooks.info

9. Enter your credentials in the Windows Security pop-up, and select OK. You may receive an additional pop-up window, as shown in Figure 8-32. Click Yes to continue.

10. Read through the details and select Yes. Your connection to the Web Role is completed.

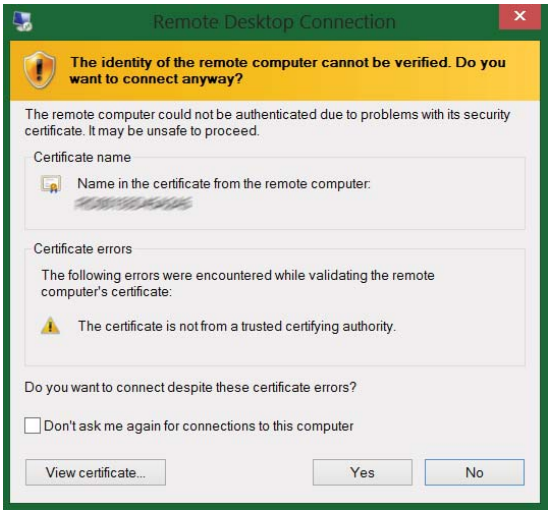

Remote Desktop Connection
The identity of the remote computer cannot be verified. Do you want to connect anyway?
The remote computer could not be authenticated due to problems with its security certificate. It may be unsafe to proceed.
Certificate name
Name in the certificate from the remote computer:
Certificate errors
The following errors were encountered while validating the remote computer's certificate:
The certificate is not from a trusted certifying authority.
Do you want to connect despite these certificate errors?
Don't ask me again for connections to this computer
View certificate...
Yes
No

FIGURE 8-32

NOTE *After you have a Remote Desktop Connection to your Cloud Server, the door is open for you to make almost any change, and look at almost any configuration, log, or performance counter. It's the same as if you were managing your own Windows Server running IIS.*

---

11. Right-click the task bar to get a pop-up menu, and select Task Manager, as shown in Figure 8-33.

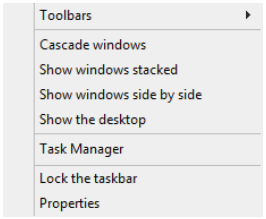


FIGURE 8-33

12. Navigate through the Task Manager looking on the Performance tab, as shown in Figure 8-34. You can see the utilization of CPU, Memory, and Network. There is a link to open the Resource Monitor as well.

www.it-ebooks.info

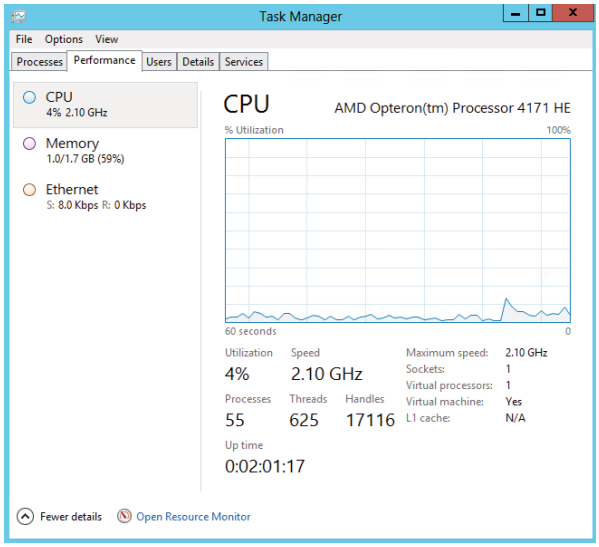


FIGURE 8-34

13. Click the Details tab, shown in Figure 8-35. Here, you see the W3WP worker process IIS uses to manage the requests being made to that Web Role. If you need to perform some debugging of a hung, crashed, or failing website/Web Role, you can right-click the W3WP worker process and select Create Memory Dump. Then you can download the memory dump and analyze it using WinDbg or Visual Studio at a later time. Figure 8-36 shows this pop-up window and the resulting message after the memory dump is created. Take note of the path so that you know where to copy it from.

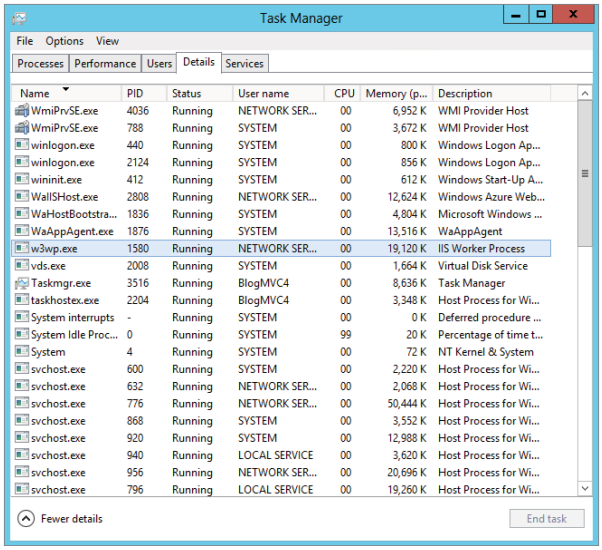

Task Manager
File Options View
Processes Performance Users Details Services
Name PID Status User name CPU Memory (p... Description
WmiPrvSE.exe 4036 Running NETWORK SER... 00 6,952 K WMI Provider Host
WmiPrvSE.exe 788 Running SYSTEM 00 3,672 K WMI Provider Host
winlogon.exe 440 Running SYSTEM 00 800 K Windows Logon Ap...
winlogon.exe 2124 Running SYSTEM 00 856 K Windows Logon Ap...
wininit.exe 412 Running SYSTEM 00 612 K Windows Start-Up A...
WaIISHost.exe 2808 Running NETWORK SER... 00 12,624 K Windows Azure Web...
WaHostBootstra... 1836 Running SYSTEM 00 4,804 K Microsoft Windows ...
WaAppAgent.exe 1876 Running SYSTEM 00 13,516 K WaAppAgent
w3wp.exe 1580 Running NETWORK SER... 00 19,120 K IIS Worker Process
vds.exe 2008 Running SYSTEM 00 1,664 K Virtual Disk Service
Taskmgr.exe 3516 Running BlogMVC4 00 8,636 K Task Manager
taskhostex.exe 2204 Running BlogMVC4 00 3,348 K Host Process for Wi...
System interrupts - Running SYSTEM 00 0 K Deferred procedure ...
System Idle Proc... 0 Running SYSTEM 99 20 K Percentage of time t...
System 4 Running SYSTEM 00 72 K NT Kernel & System
svchost.exe 600 Running SYSTEM 00 2,220 K Host Process for Wi...
svchost.exe 632 Running NETWORK SER... 00 2,068 K Host Process for Wi...
svchost.exe 776 Running NETWORK SER... 00 50,444 K Host Process for Wi...
svchost.exe 868 Running SYSTEM 00 3,552 K Host Process for Wi...
svchost.exe 920 Running SYSTEM 00 12,988 K Host Process for Wi...
svchost.exe 940 Running LOCAL SERVICE 00 3,620 K Host Process for Wi...
svchost.exe 956 Running NETWORK SER... 00 20,696 K Host Process for Wi...
svchost.exe 796 Running LOCAL SERVICE 00 19,260 K Host Process for Wi...
Fewer details
End task

FIGURE 8-35

www.it-ebooks.info

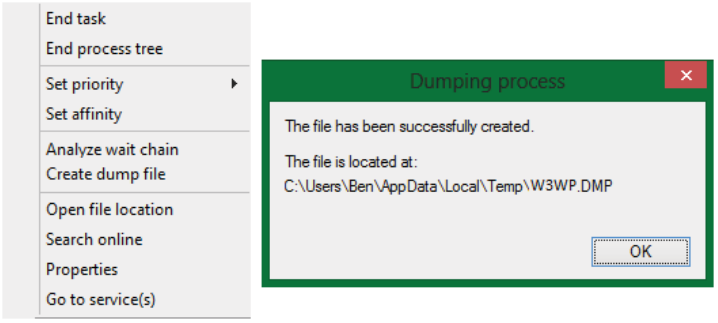


FIGURE 8-36

14. Open the IIS management console by selecting Tools from the Server Manager Dashboard ➪ Internet Information Services (IIS) Manager, as shown in Figure 8-37.

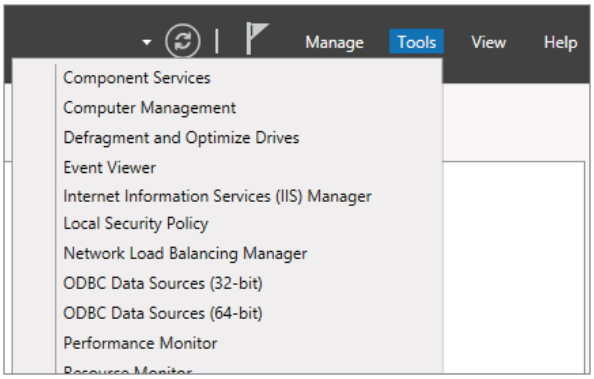


FIGURE 8-37

15. Many other tools are available for access, usage, and review. For example, after IIS is open, if you click the Worker Processes icon (shown in Figure 8-38), this displays the requests currently executed on your Cloud Service in real time.

NOTE *A worker process is the W3WP.exe windows process. It handles requests sent to a web server for a specific application pool.*

---

www.it-ebooks.info

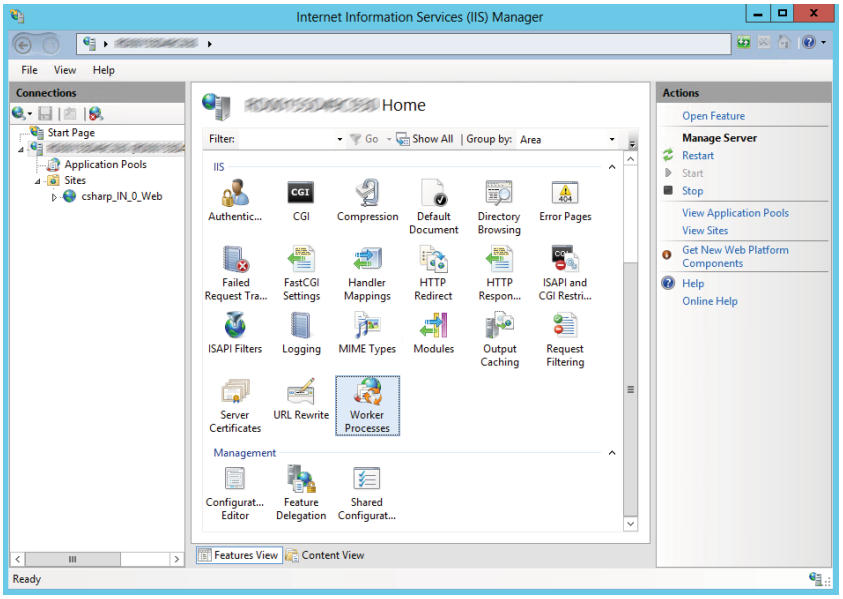


FIGURE 8-38

Figure 8-39 represents a view of the currently executing processes. The information provided is the amount of CPU utilization, memory utilization, and the process ID handling the given request.

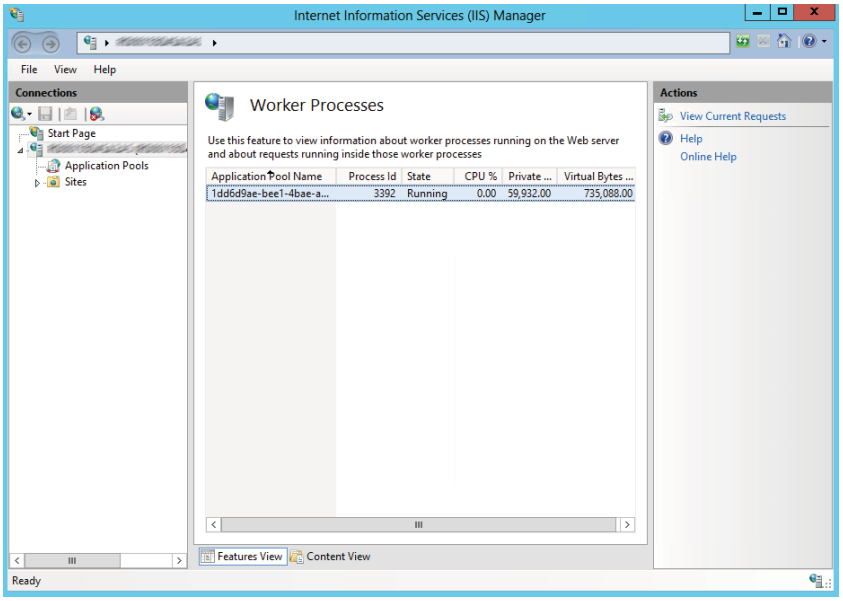


FIGURE 8-39

www.it-ebooks.info

You can double-click the specific request for even more information:

- **Website ID:** A unique identification number given to the website
- **URL:** The Internet address of the request
- **Verb:** Example: GET, POST, PUT, HEAD, and so on
- **Client IP:** The IP address of the client that sent the request to the server
- **State:** `ExecuteRequestHandler`, `BeginRequest`, and more
- **Module Name:** `ManagedPipelineHandler`, `IsapiModule`, and such
- **Time Elapsed:** The amount of time required to execute the request

These features provide some very useful information for troubleshooting and for learning how your system works on the Windows Azure and IIS platform.

Another tool available to monitor and administer your Web Role in IIS on the Windows Azure platform is the PowerShell WebAdministration module. This provides the administrator with, among others, the ability to:

- Add, modify, and delete websites and application pools
- Enable and disable global modules
- Back-up and restore a web configuration

Using the WebAdministration PowerShell Cmdlets

To use the WebAdministration components of PowerShell that are useful for administering and monitoring IIS, perform the following steps. PowerShell uses cmdlets, which are small programs that perform a specific actions; for example, Get-WebRequest is a cmdlet.

1. To Start PowerShell click the PowerShell icon on the left side of the task bar, as shown in Figure 8-40.

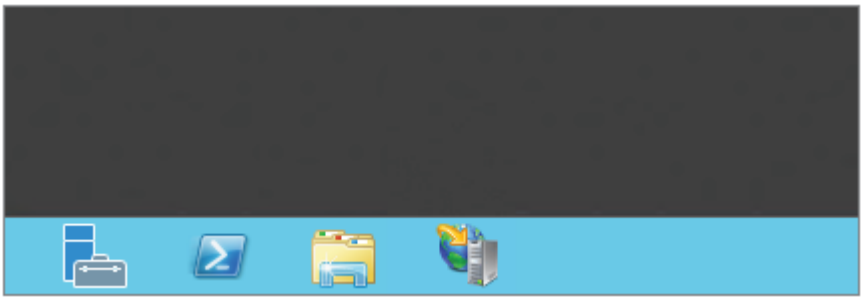


FIGURE 8-40

www.it-ebooks.info

2. After the command window opens, enter the command **`import-module WebAdministration`** ➪ **`cd IIS:`** ➪ **`cd sites`** ➪ **`dir`** to get a list of the websites running on this instance of IIS. Figure 8-41 illustrates how this may appear.

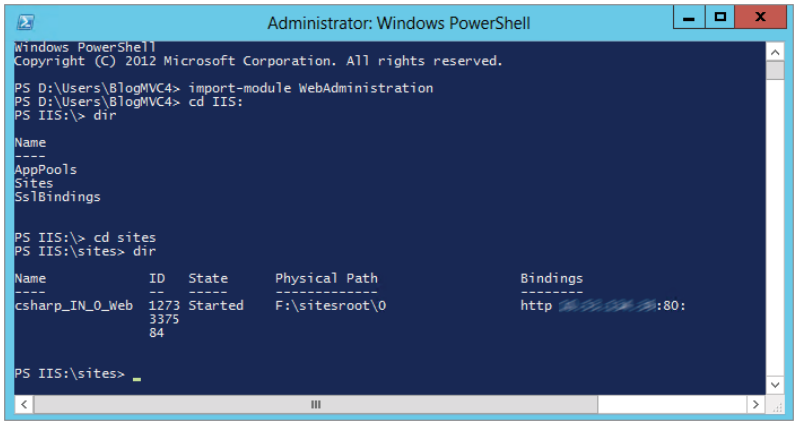


FIGURE 8-41

3. You can see if the website is in a started or stopped state. You can also check the same for application pools running on the server. Enter **`cd`** ➪ **`cd AppPools`** ➪ **`dir`** to get a list and status of the application pools, as shown in Figure 8-42.

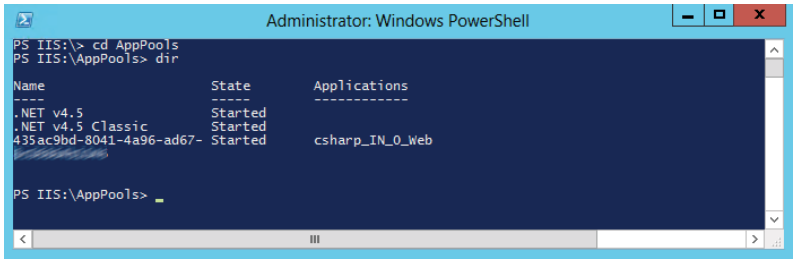


FIGURE 8-42

4. You can also gain some information about currently executing requests by entering **GET** ➪ **`WebRequest`** ➪ **`AppPool`** ***GUID***, as shown in Figure 8-43.

The result of the command provides you information about a request that the Web Server is currently responding to. Information includes an ID, the requested page, the verb, the hostname, and so on, and is useful for obtaining a real-time snapshot of what is happening on the system. For example, if users are complaining about some performance problems, you can execute this command to see if anything looks suspicious, like a large `timeElapsed` value.

www.it-ebooks.info

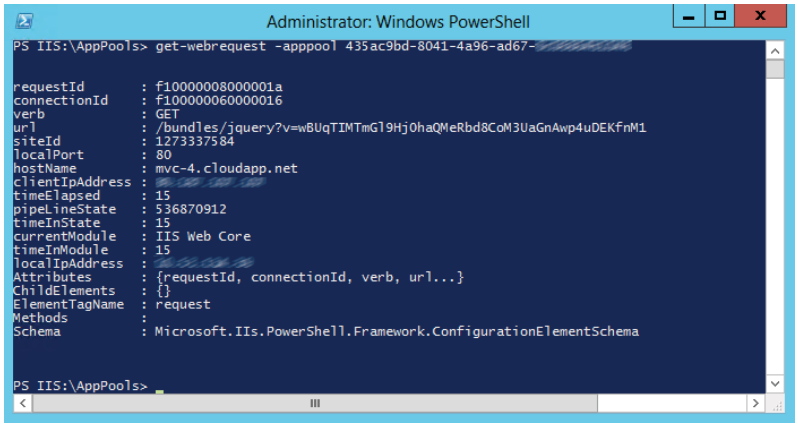


FIGURE 8-43

Changing IIS Settings Using the IIS Management Console

You can change IIS settings to improve the performance and stability of the website using the IIS management console. To do so, follow these steps:

1. Open the IIS management console, and click the Logging feature, as shown in Figure 8-44, to view where the IIS logs are stored.

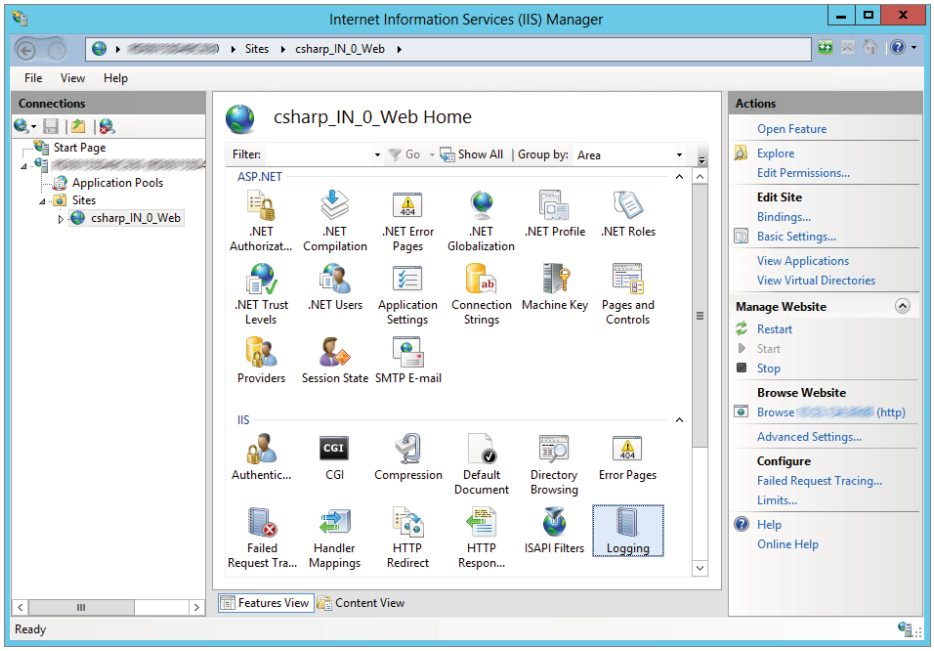


FIGURE 8-44

www.it-ebooks.info

2. The path in the Directory text box, shown in Figure 8-45, is the location to which the IIS logs are written. Open Windows Explorer and navigate to that path to view the IIS Log files.

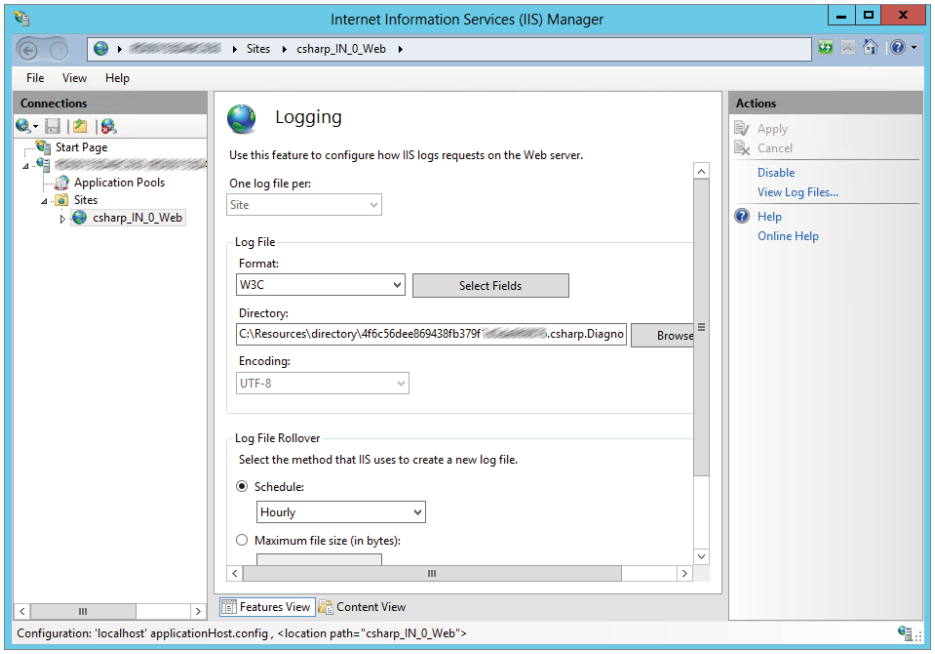


FIGURE 8-45

3. Review the ISS logs. You use these Log files to troubleshoot any other IIS issue and draw a lot of conclusions and actions by reviewing them.

NOTE *You can find additional resources for troubleshooting IIS issues using IIS Logs in an article the author wrote about troubleshooting IIS issues using LogParser here:* `http://www.iis.net/learn/troubleshoot/performance-issues/troubleshooting-iis-performance-issues-or-application-errors-using-logparser`.

To illustrate how to use log files to troubleshoot, I use Listing 8-4 as an example. After downloading and installing LogParser, open a command prompt and execute the query using LogParser, as shown in Listing 8-4.

www.it-ebooks.info

LISTING 8-4: LogParser Query — General Activity

```
logparser.exe "SELECT sc-status, sc-substatus, COUNT(*)
               FROM *.log
               GROUP BY sc-status, sc-substatus
               ORDER BY sc-status" -i:w3c
```

This query renders results similar to those shown in Table 8-3. The query and its output provide some insight into what is happening on the website. The large majority of requests handled by the website/Web Role are successful, that is, a request with a sc-status of 200 is considered successful. 300 and 400 HTTP Status codes can be "generally" assumed to be okay, unless you have an authentication problem. Otherwise, the most important HTTP Status codes to analyze further are requests that result in an HTTP Status code of 500.

TABLE 8-3: Results of an example LogParser query executed on IIS Logs by Status

SC-STATUS	SC-SUBSTATUS	COUNT(*)
200	0	920349
301	0	1031
304	0	78705
401	2	92006
404	0	2935
500	0	4187

An HTTP Status code of 500 means there was some kind of error on the server that prevented the successful response to the request. To investigate the 500 HTTP Status codes further, you execute the LogParser query shown in Listing 8-5.

LISTING 8-5: LogParser Query — 500 HTTP Status Error Deep Dive

```
logparser.exe "SELECT cs-uri-stem, COUNT(*)
               FROM *.log
               WHERE sc-status=500
               GROUP BY cs-uri-stem
               ORDER BY COUNT(*) DESC" -i:w3c
```

The LogParser query renders the requested file that resulted in a 500 HTTP Request and the number of time it happened for that file. Table 8-4 illustrates an example of what the results may look like.

www.it-ebooks.info

TABLE 8-4: Results of an example LogParser query executed on IIS Logs by File

CS-URI-STEM	COUNT(*)
Default.aspx	1568
GetDetails.aspx	315
ViewOrders.aspx	53
. . .	. . .

The result of the query clearly shows that the file with the most failures is the Default.aspx file. Therefore, you need to look into the contents of the file to determine what it is doing and the dependencies it has, and then take some actions to make it more fault-tolerant. Using this information together with other available logs, you can compose a good theory of what is happening; for example, viewing the Application event logs existing in the Event Viewer, as discussed next.

Viewing the Event Logs in a Windows Azure Cloud Service

You can also view and configure event logs via the Event Viewer window. The event logs contain system and application errors and warnings that are very useful in troubleshooting and preventing system issues. To view the logs, perform the following:

1. Select Event Viewer, as shown in the previous figure, and the Event Viewer window displays, as shown in Figure 8-46.

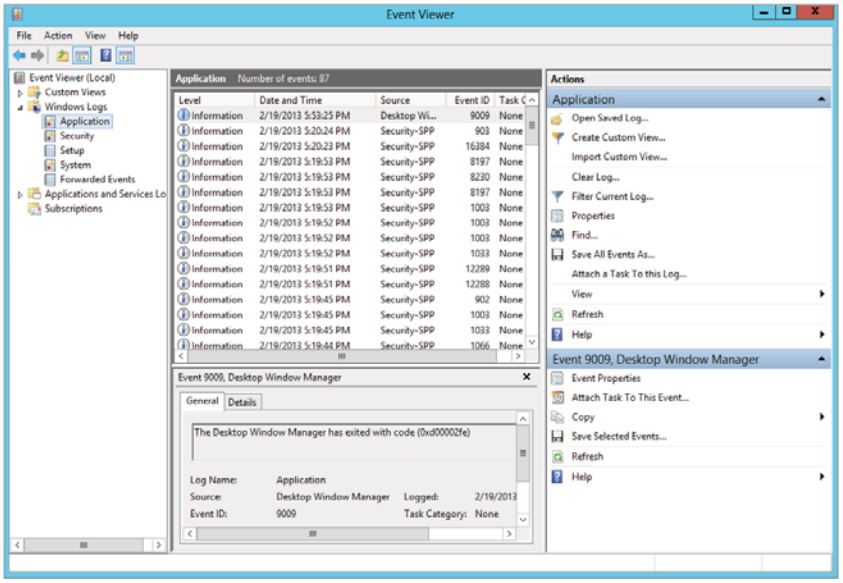

Event Viewer
File Action View Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Lo
Subscriptions
Application Number of events: 87
Level Date and Time Source Event ID Task C
Information 2/19/2013 5:53:25 PM Desktop Wi... 9009 None
Information 2/19/2013 5:20:24 PM Security-SPP 903 None
Information 2/19/2013 5:20:23 PM Security-SPP 16384 None
Information 2/19/2013 5:19:53 PM Security-SPP 8197 None
Information 2/19/2013 5:19:53 PM Security-SPP 8230 None
Information 2/19/2013 5:19:53 PM Security-SPP 8197 None
Information 2/19/2013 5:19:53 PM Security-SPP 1003 None
Information 2/19/2013 5:19:52 PM Security-SPP 1003 None
Information 2/19/2013 5:19:52 PM Security-SPP 1003 None
Information 2/19/2013 5:19:52 PM Security-SPP 1033 None
Information 2/19/2013 5:19:51 PM Security-SPP 12289 None
Information 2/19/2013 5:19:51 PM Security-SPP 12288 None
Information 2/19/2013 5:19:45 PM Security-SPP 902 None
Information 2/19/2013 5:19:45 PM Security-SPP 1003 None
Information 2/19/2013 5:19:45 PM Security-SPP 1033 None
Information 2/19/2013 5:19:44 PM Security-SPP 1066 None
Event 9009, Desktop Window Manager
General Details
The Desktop Window Manager has exited with code (0xd00002fe)
Log Name: Application
Source: Desktop Window Manager Logged: 2/19/2013
Event ID: 9009 Task Category: None
Actions
Application
Open Saved Log...
Create Custom View...
Import Custom View...
Clear Log...
Filter Current Log...
Properties
Find...
Save All Events As...
Attach a Task To this Log...
View
Refresh
Help
Event 9009, Desktop Window Manager
Event Properties
Attach Task To This Event...
Copy
Save Selected Events...
Refresh
Help

FIGURE 8-46

www.it-ebooks.info

NOTE *In most cases, IIS, ASP.NET, and ASP.NET MVC logs are logged to the Application Event log located in the Event Viewer (local) ➪ Windows Logs ➪ Application log path.*

2. Set a Filter to view only Errors or Warnings and see if you find any troubles with your system or the platform.

The Event Viewer offers many additional options for logging and troubleshooting a problem. For example, if you have some problems with a certificate, enable the CAPI2 Application Log, as shown in Figure 8-47, to log Errors, Warnings, and Information pertaining to SSL certificate issues.

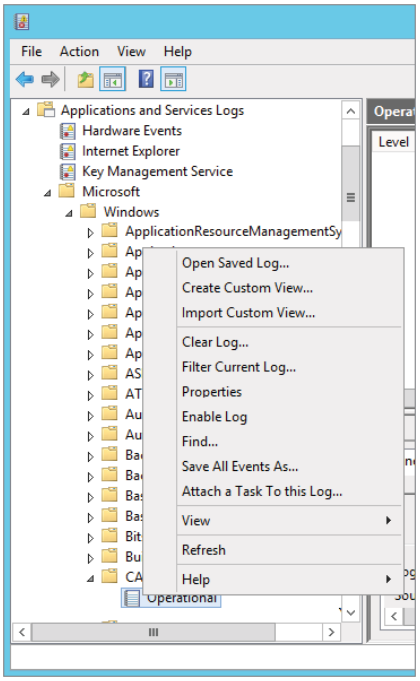


FIGURE 8-47

Viewing the Cloud Service Usage Dashboard

Moving back to the Windows Azure management console, a useful Dashboard can help you track the CPU utilization and a number of other metrics, as shown in Figure 8-48. Tracking these metrics is important for setting baselines and forecasting future resource requirements.

www.it-ebooks.info

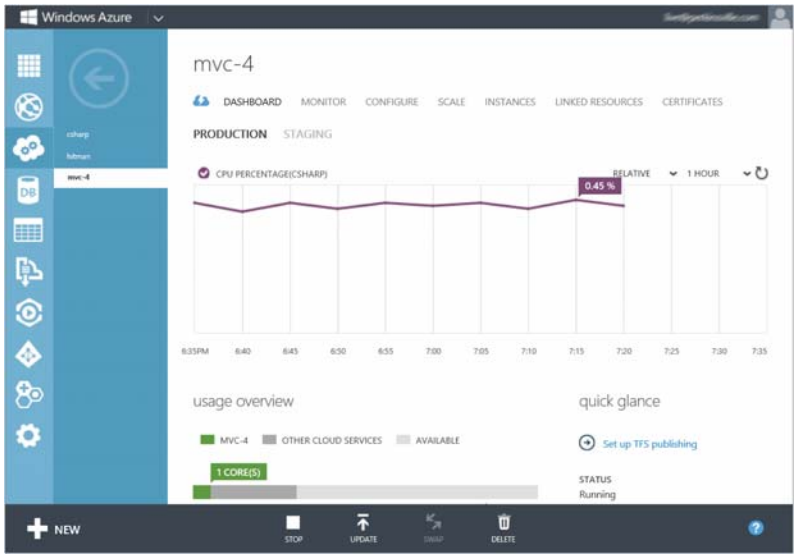


FIGURE 8-48

To get a good overview understanding of how use these monitoring features, perform the following steps:

1. Click the Monitor link at the top of the management console to access the page shown in Figure 8-49. Notice the Add Metrics link at the bottom of the page.

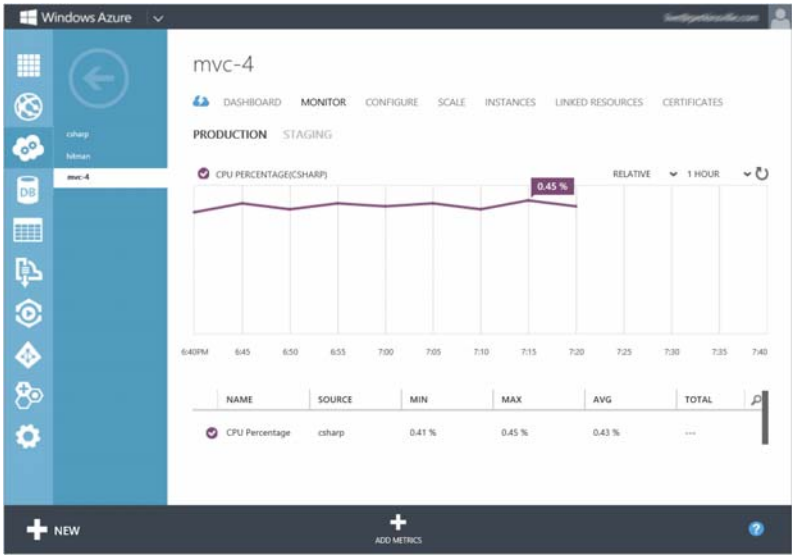


FIGURE 8-49

www.it-ebooks.info

2. Click the Add Metrics link and the pop-up shown in Figure 8-50 displays.
3. Select the properties you want to monitor; then click the check on the lower-right of the window. The metrics you selected in Figure 8-50 are added.

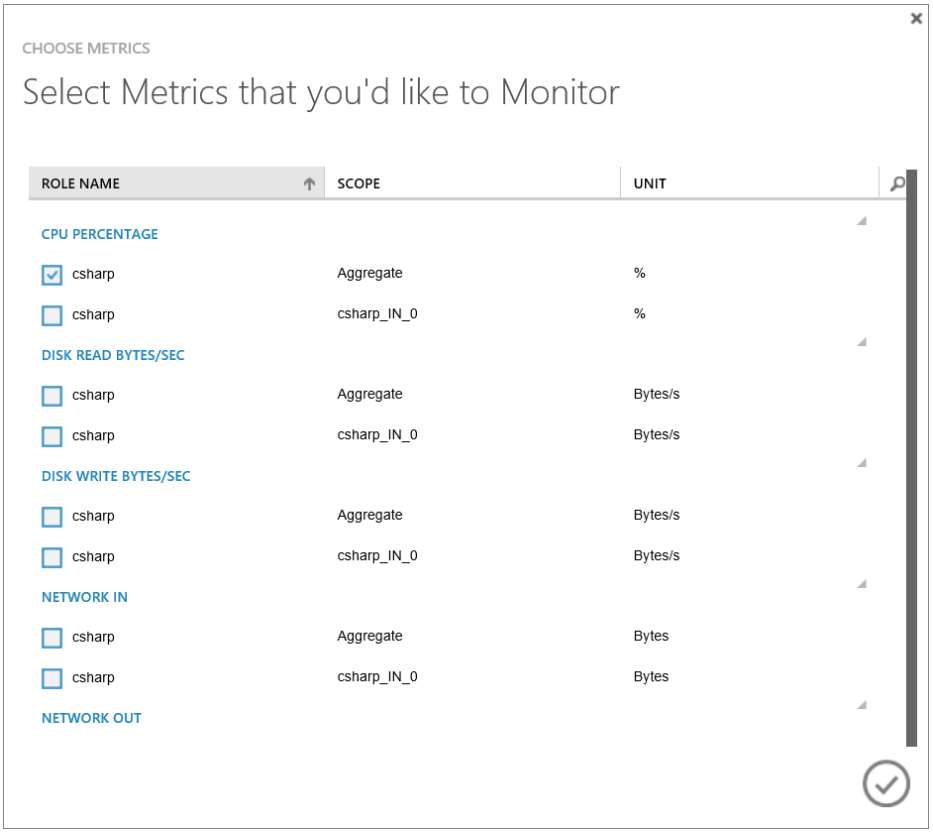


FIGURE 8-50

4. In the Windows Azure management console for the csharp Web Role, select the Configure link at the top of the page. Figure 8-51 shows this page.
5. Configure the monitoring level to Verbose and keep the Retention Days as the default value. When selected, the menu items at the bottom of the page change, as represented in Figure 8-52. A warning appears.

www.it-ebooks.info

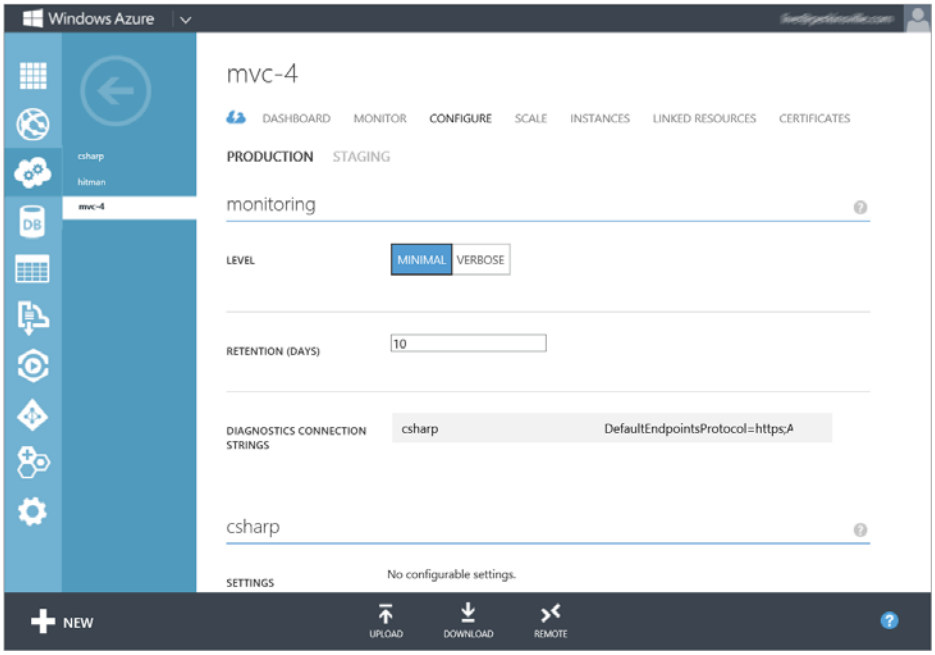


FIGURE 8-51

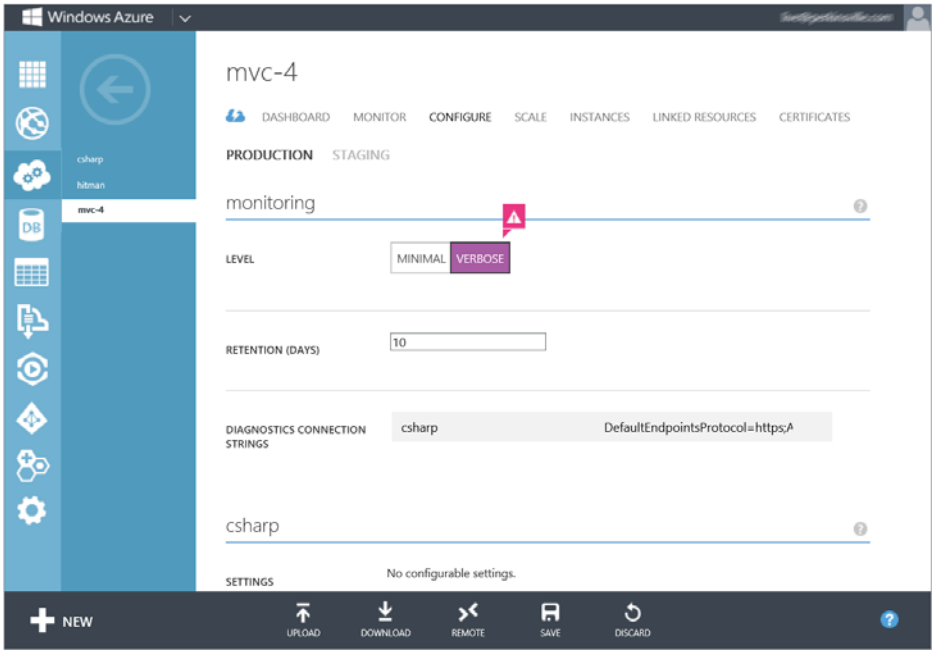


FIGURE 8-52

www.it-ebooks.info

WARNING *Regarding the warning shown in Figure 8-52: Changing the level of monitoring to Verbose can have an impact on the costs, so make sure you take this into consideration when making the change.*

7. Select the Save button to apply the changes.

SUMMARY

In this chapter you learned that there are some large differences in the monitoring and support capabilities between a website and a Cloud Service. The biggest difference is that a Windows Azure Cloud Service supports Remote Desktop Connection, something not possible with a Windows Azure Web Site.

Using a Remote Desktop Connection, you can administer the Web Role just like you would any website hosted on IIS. All the tools for monitoring the behavior of the website and operating system, such as Performance Monitor, Event Viewer, Task Manager, and so on are at your fingertips.

Nonetheless, plenty of capabilities and features are available for applications hosted as a website, such as usage statistics on the Dashboard, the capability to add metrics to the Dashboard, and the quick availability of download log files, such as the IIS logs and Failed Request Tracing logs. All these features can provide you with enough information to understand and resolve most issues experienced on a Windows Azure Web Site.

www.it-ebooks.info